

Universitat de Lleida

Escola Politècnica Superior

Enginyeria Tècnica en Informàtica de Sistemes

Treball de final de carrera

**Estudi i implementació d'un simulador de cicle tancat
de producció de porcs.**

Autor: Carlos Miquel Fernàndez

Director: Lluís Miquel Pla Aragonés

14-7-2009

índex

1.- INTRODUCCIÓ. 5

1.1.- Objectius. 6

2.- DESCRIPCIÓ DEL SISTEMA REAL. 10

2.1.- Introducció a les explotacions de cria. 10

2.2.- Descripció del cicle de vida d'un animal. 12

3.- CARACTERÍSTIQUES DE L'APLICACIÓ. 16

3.1 Simulació discreta de sistemes 16

3.2.- Anàlisi de Requeriments Funcionals. 17

3.1.1.- Introducció de formules de la producció porcina 18

3.1.2.- Càlcul en períodes dinàmics sobre dades simulades.18

3.1.3.- Generació d'un codi general que tingui les dades a introduir.19

3.1.4.- Introducció de les dades obtingudes.19

3.1.5.- Presa de decisió en el bloc de l'engreix.19

3.1.6.- Validació dels ensenyaments amb dades reals.20

3.3.- Anàlisi de Requeriments NO Funcionals. 20

3.4 Implementació 21

3.5. Dificultats en la implementació 24

3.6.- Recursos del Sistema. 25

3.6.1.- Recursos Hardware. 25

3.6.2.- Recursos Software. 26

3.6.3.- Restriccions tècniques. 26

4.- PLANIFICACIÓ DEL PROJECTE. 28

4.1.- Identificació i Gestió de Riscs. 28

4.1.1.- Identificació dels riscs. 28

4.1.2.- Projecció dels riscs. 30

4.1.3.- Gestió de riscs. 31

4.2.- Planificació. 33

5.- PROVES DE VALIDACIÓ DEL SISTEMA. 36

5.1.- Prova d'Unitat. 36

5.2.- Prova d'Integració. 37

5.3.- Prova de Validació. 38

5.4.- Prova de Sistema. 38

5.4.1.- Prova de Recuperació. 39

5.4.2.- Prova de Seguretat. 39

5.4.3.- Prova de Resistència. 39

5.4.4.- Prova de rendiment. 40

6.- GARANTIA DE QUALITAT DEL SOFTWARE. 41

6.1.- Característiques Operatives. 42

6.2.- Capacitat d'Adaptació a d'altres Entorns. 43

6.3.- Capacitat d'Adaptació a canvis. 43

7.- CONCLUSIONS. 45

BIBLIOGRAFIA. 48

Annex I 49

Annex II 56

Annex III 63

ÍNDEX DE FIGURES.

Figura 1. Esquema general de les parts o objectius del projecte. 7

Figura 2: Relació entre cost i garrins aconseguits per truja/any – (Becana 1998). 10

Figura 3: Cicle de vida d'un animal. 12

Figura 4: Diagrama dels estats de cada cicle. 14

Figura 5: Taula de riscos i ponderacions 30

Figura 6: Taula de tasques del projecte 33

1.- INTRODUCCIÓ.

Els responsables de les explotacions porcines busquen el màxim rendiment de les seves granges amb el mínim cost possible. Sempre, això requereix l'ús d'eines de gestió a les explotacions porcines.

Avui en dia, s'utilitza la informàtica per tal d'ajudar al ramader en la seva gestió, i les eines que utilitzen els ordinadors evolucionen cap a noves funcionalitats, facilitats d'ús i mètodes per a l'ajut a la presa de decisions directament o a través de consulta. En aquest context, el Departament de Matemàtica, Grup de model de decisions, computació i simulació, disposa d'un conjunt d'eines informàtiques d'ajuda a la presa de decisions per tal de controlar la gestió de diferents explotacions de producció porcina. D'aquesta manera s'aconsegueix millorar les estratègies del maneig productiu.

La finalitat d'aquest projecte és l'estudi i posterior implementació d'una eina visual desenvolupada amb el llenguatge de programació C i amb l'eina del simulador *Extend*, que simuli el pas del temps en una granja dedicada a la producció de garrins i permeti realitzar un anàlisi prospectiu de l'activitat.

L'aplicació que es pretén desenvolupar afegeix noves funcionalitats a una eina ja existent la qual captura la informació referent als esdeveniments que es van produint entre la població d'una granja determinada i en calcula diferents paràmetres i índexs. Aquests paràmetres serveixen d'entrada per a un simulador de la granja individual. Els resultats del model serveixen per conèixer millor les conseqüències de la gestió que es porta a terme a la granja.

En essència, la nova funcionalitat a afegir consistirà en simular les truges individualment i el pas del temps. La forma de fer-ho és introduint dades preses de les granges de garrins al simulador. Aquest simularà el cicle de vida dels garrins i proporcionarà informació als usuaris respecte l'estat de la granja en un futur si es segueix una política determinada, ja sigui a curt o a llarg termini.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Aquest tipus de sistemes informàtics reben el nom d' "eines d'ajuda a la presa de decisions", DSS en anglès, *Decision Support Systems*, ja que en certa manera, l'eina a desenvolupar servirà per explorar diferents alternatives de maneig i triar quina és millor per tal de maximitzar beneficis i minimitzar costos.

Per tal de garantir la garantia de qualitat de l'eina desenvolupada, el desenvolupament del projecte s'estructura en les les fases següents:

- Recollida de requeriments.* En aquesta fase es produeix una primera presa de contacte entre l'alumne desenvolupador i el professor. El que es pretén és que l'alumne es faci una primera idea dels requeriments que l'eina a desenvolupar necessita.
- Anàlisi.* En aquesta fase s'estudien els requeriments anteriors per tal de comprendre amb tot detall quina ha de ser la funcionalitat de l'eina.
- Disseny.* A partir d'aquesta fase es prepara el disseny de l'eina. D'aquesta manera s'aconsegueix major facilitat a l'hora d'implementar i s'obté una més gran flexibilitat si ca, en un futur, realitzar canvis per causes no previstes.
- Implementació.* En aquesta fase es tradueix a codi C i *Extend* tot el disseny realitzat en la fase anterior. Depenent del disseny, aquesta fase serà més o menys fàcil i ràpida d'implementar.
- Proves.* Si partim de la base de que tota aplicació desenvolupada conté errors, aquesta fase ens permetrà localitzar aquests errors i solucionar-los. La utilitat final és la de localitzar i solucionar el major nombre d'errors possibles, ja que fer-ho en la seva totalitat és impossible.

Cada fase serà estudiada amb més detall en successius apartats de la documentació.

1.1.- Objectius.

El projecte ha estat dividit en 5 etapes diferenciades:

1. *Millora de la complexitat del model incluint nous paràmetres*
2. *Automatització de l'anàlisi de paràmetres extrems de granges reals.*
3. *Automatització de la introducció de dades obtingudes del previ pre-càlcul.*
4. *Creació del bloc d'Engreix i de Transició.*
5. *Validació dels ensenyaments amb dades reals.*

Tot seguit passem a detallar cadascuna de les etapes.

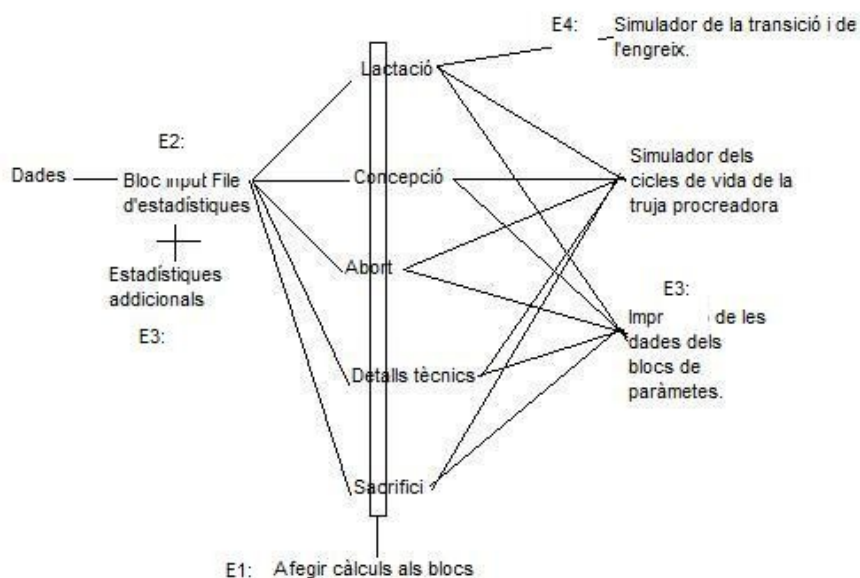


Figura 1. Esquema general de les parts o objectius del projecte.

Primera etapa: Millorar la complexitat del model incloent nous paràmetres

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

L'objectiu d'aquesta primera etapa és el d'implementar una biblioteca de funcions que proporcionen millores al càlcul.

Aquesta etapa correspon a E1 de la figura anterior.

Segona etapa: Automatitzar l'anàlisi de paràmetres extrems de granges reals.

L'objectiu d'aquesta segona etapa és el de crear una biblioteca de funcions que proporcionen càlculs estadístics sobre dades del cicle de vida de les truges.

Aquesta etapa correspon a E2 de la figura anterior.

Tercera etapa: Automatització de la introducció de dades obtingudes del previ pre-càlcul.

L'objectiu d'aquesta tercera etapa és el de crear una aplicació dins d'un bloc de manera que automatitzi tota la feina anterior i introdueixi les dades obtingudes als paràmetres d'entrada del simulador.

Aquesta etapa correspon a E3 de la figura anterior.

Quarta etapa: Creació del bloc d'Engreix i de Transició.

L'objectiu d'aquesta quarta etapa és el de crear un nou simulador dels processos d'engreix i transició. Per fer això es dissenyarà una biblioteca de funcions que proporcionaran el temps de transició de cada etapa.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Aquesta etapa correspon a E4 de la figura anterior.

Cinquena etapa: Validació dels ensenyaments amb dades reals.

L'objectiu d'aquesta cinquena etapa és el de fer un procés de perfeccionament i millora o afinament de totes les etapes anteriors validant els seus resultats.

A més a més, els objectius secundaris que s'han realitzat en la implementació són:

- 1-Familiarització amb el llenguatge modl i el programa *ExtendSim* 6
- 2-Maneig amb el llenguatge c per fer programes d'estadístiques
- 3-Creació de la llibreria en dll i programació en el llenguatge modl i fer els blocs de input i output de dades.
- 4-Creació de blocs i programar-los.
- 5-Validar les dades.

2.- DESCRIPCIÓ DEL SISTEMA REAL.

Per simular els cicles de vida d'explotacions porcines de cria és necessari conèixer com funciona realment el sistema productiu en una granja. En aquest capítol s'explica el funcionament real d'una granja de producció de garrins i també, de l'engreix. D'aquesta manera es pretén que s'entengui millor *què* es tracta de simular.

2.1.- Introducció a les explotacions de cria.

Les explotacions es poden agrupar depenent de l'activitat que en elles es realitza. Distingirem tres tipus d'explotacions:

- Granja de cicle tancat o cicle complet de cria i engreixament.
- Granja de cria.
- Granja d'engreixament.

Les explotacions de producció de garrins son les que produeixen, a partir de truges mares, garrins. El destí d'aquests serà la pròpia granja, per acabar convertint-se en animals productius, o bé seran venuts a altres explotacions. La rendibilitat d'aquest tipus d'explotacions ve donat per el número de garrins aconseguits per truja i any, de forma que les despeses de cada truja es distribuirà entre més o menys depenent de la quantitat

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

de garrins venuts. Segons això, el cost de producció d'un garrí en funció de la producció per truja/any serà aproximadament:

Figura 2: Relació entre cost en euros i garrins aconseguits per truja/any – (Becana 1998).

Com es pot apreciar a la figura, el cost de producció d'un garrí augmenta a mesura que disminueix la producció per truja/any i per això el benefici disminueix.

La producció de garrins es dona dins el cicle de vida de les mares. Aquest cicle de vida i les seves etapes s'explica amb tot detall al següent apartat.

2.2.- Descripció del cicle de vida d'un animal.

Després d'explicar el cicle de vida d'un animal reproductor, ens serà més senzill entendre què és exactament el que volem simular. La part del projecte que engloba el simulador, tractarà de simular amb tot detall el comportament del cicle de vida d'un animal.

Es tracta de crear la generació de blocs que simulin en cada una de les etapes del cicle de vida, seguint un ordre cronològic que ens permeti creure que les dades simulades formen part d'una explotació de producció de garrins real.

A més, l'eina simuladora també tindrà en compte la possibilitat de que l'animal reproductor causi baixa mentre aquest es trobi en qualsevol estat del cicle. D'aquesta manera podem simular el cas de que l'animal es posi malalt o sigui donat de baixa per altres raons. Així doncs s'aconseguirà un major grau de realisme en l'aplicació.

El model que actualment volem simular ens ve condicionat pels cicles reproductius de les mares productores de garrins.

Cada animal reproductor passa per diferents etapes, que venen a ser un conjunt finit de cicles productius. Cada cop que acaba un cicle se'n comença un de nou. Aquest procés es repetirà fins el moment de la mort o baixa de l'animal.

Naixement o Compra



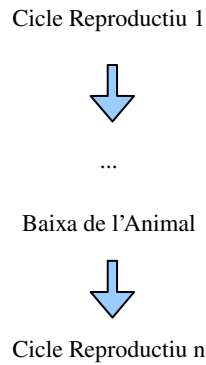


Figura 3: Cicle de vida d'un animal.

Cada cicle disposa d'una estructura bàsica d'estats, que seran comuns a tots els cicles i que els animals aniran repetint cada cop, tot i que els paràmetres seran diferents si per exemple les introduïm des d'un fitxer d'entrada. Aquest conjunt d'estats es mostren gràficament a la figura 5, on mostrem el cicle reproductiu n.

La manera correcta de llegir aquesta gràfica és de dalt a baix i d'esquerra a dreta. D'aquesta manera entendrem el conjunt total d'etapes que ha de passar cada animal dins d'un cicle.

El cicle s'inicia quan l'animal ha acabat el cicle anterior (cicle n-1) o és donat d'alta a l'explotació. En tot cas l'animal es troba a l'edat adulta i ja està preparat per produir garrins. A partir d'aquí comencen les etapes de **cobriment**. L'etapa de cobriment és el període de temps en que l'animal està en espera de ser cobert un cop s'aprecia el zel.

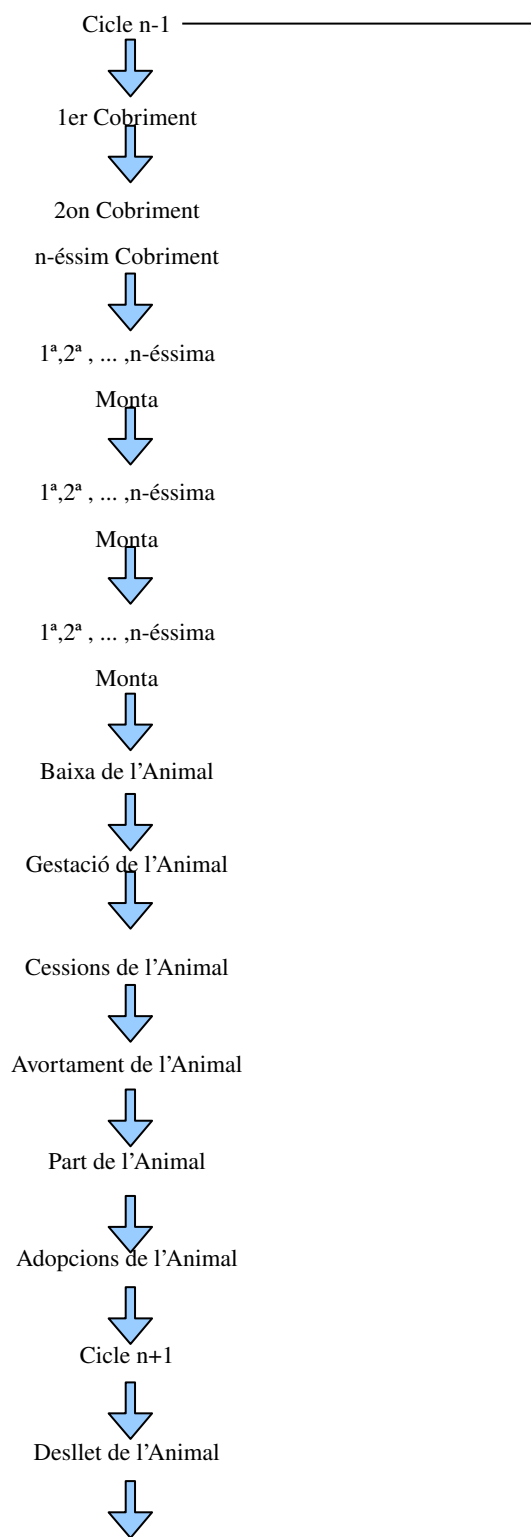


Figura 4: Diagrama dels estats de cada cicle.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Cada cobriment, representa un cicle estral de l'animal. En cada cicle estral es **monta** l'animal, ja sigui natural o artificialment, per tal de que aquest quedi prenyada. Actualment el 100% solen ser IA. Si l'animal no quedés prenyat, es passaria a un següent cobriment en el que es tornaria a intentar prenyar l'animal. Es realitzaria aquesta operació un número limitat de vegades. Si l'animal no quedés prenyat en aquests n intents, es donaria de **baixa** de l'explotació i la seva plaça seria ocupada per un altre animal.

Un cop l'animal ha quedat prenyat s'inicia l'etapa de **gestació**. En qualsevol estat, l'animal pot causar baixa, ja sigui per malaltia o per desig del responsable de l'explotació. Aquest concepte no està reflectit al diagrama però s'ha de tenir en compte en tot moment. Un motiu per que el responsable doni de baixa una truja potser una gestació fallida, és a dir, un avortament. També perquè la truja no quediprenyada i calgui repetir la IA. Si es produeix un avortament, l'animal és donat de baixa de l'explotació i tot seguit és reemplaçat.

En canvi, si la gestació es desenvolupa amb normalitat, al final de la mateixa es produeix un **part**. En aquest cas, l'animal obté un número de garrins vius determinat. Amb això tenim que aquests garrins poden ser cedits a altres mares per alguna circumstància de la cria o, també, el mateix animal pot adoptar garrins d'altres mares per les mateixes circumstàncies. En el cas del projecte actual, l'eina simuladora no té present aquestes possibles adopcions – cessions, amb el que considera que tots els animals que neixen d'una mare es queden amb ella fins al desllet

Un cop s'ha produït el part, els garrins nascuts segueixen el seu procés de cria junt amb la seva mare fins que aquests són prou grans. A partir d'aquest moment es produeix el

desllet. Això significa que es separen els garrins de la mare i passen a formar part d'un lot d'animals de cria que compleixen els seus creixements fins arribar a ser venuts a l'escorxador o ser venuts a altra o bé integrar-se com a reproductors.

3.- CARACTERÍSTIQUES DE L'APLICACIÓ.

3.1 Simulació discreta de sistemes.

La natura del sistemes de simulació tracta d'un conjunt de tècniques que serveixen per tal d'imitar, simular les operacions de varis tipus dels processos reals. Aquest conjunt de processos s'anomena sistema i s'utilitza per estudiar-lo científicament per un conjunt d'assumpptes en què treballem. Aquests assumpptes amb els que habitualment els fem anar de forma matemàtica o de relacions lògiques, constitueixen el model que es usat per intentar entendre com funciona el comportament del sistema.

Si les relacions que componen el sistema es bastant simple, pot ser possible utilitzar diferents mètodes matemàtics (àlgebra, càlculs, etc.) per obtenir informació exacta en qüestions d'interès, de forma analítica. De totes formes, la majoria de sistemes reals són massa complexes per avaluar-los analíticament i per això s'utilitza la simulació de mitjanes estadístiques. En una simulació usem ordinadors per avaluar un model numèric, i amb dades per estimar les característiques ideals del model.

La simulació discreta es basa en l'execució d'accions realitzades de forma atzarosa durant un determinat temps degut a un esdeveniment. En el moment que es realitza l'esdeveniment s'executa una funció externa que a partir de les seves instruccions aconseguix un fi. En canvi, en la simulació contínua, el procés s'executa de forma constant.

Com a exemple de la simulació discreta considerem l'empresa de la cria de porcs. Aquest exemple intenta justificar els beneficis en productivitat en funció dels seu cost d'inversió. A part de l'estudi eficiència-cost també permet veure com afectarien noves ampliacions.

3.2.- Anàlisi de Requeriments Funcionals

Els requeriments funcionals ens proporcionen una descripció del comportament desitjat del software. Cada requeriment funcional, expressa una relació entre les entrades i sortides del sistema, és a dir, especifica les sortides que s'han de produir a partir d'unes determinades entrades i les operacions necessàries per aconseguir això.

També han d'especificar com s'ha de comportar el sistema davant de situacions anormals (entrades invàlides, errors, etc.).

En el nostre cas, podem diferenciar els següents requeriments funcionals:

- Introducció de fórmules de la producció porcina
- Anàlisi de la granja que vol ser simulada.
- Selecció de les dades rellevants.
- Introducció al simulador de les dades seleccionades.
- Presa de decisió en el bloc de l'engreix.
- Validació dels ensenyaments amb dades reals.

3.2.1.- Introducció de fórmules de la producció porcina

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Aquest primer requeriment es desenvoluparà usant codi *modL* del *ExtendSim* 6 per tal d'implementar unes petites aplicacions de càlcul sobre les dades de les granges de garrins. A aquests càlculs s'hi podrà accedir a partir de la botonera del bloc corresponent.

Aquest requeriment es pot veure com un pas preliminar a les etapes descrites al capítol 1 (objectius del treball).

3.2.2.- Anàlisi de la granja que vol ser simulada.

L'objectiu d'aquest segon requeriment és el de crear un mòdul que permetrà realitzar una sèrie de càlculs estadístics que, després, s'usaran als diferents blocs del simulador, o sigui, a les diferents etapes de vida de la truja. Com hem vist al capítol anterior, aquestes etapes són les de *lactació*, *destetat*, *engreix*, *cubició*, *gestació* i *part*. Alguns dels càlculs estadístics desenvolupats en aquest mòdul i que seran usats a les diferents etapes són: la mitjana, mediana, desviació, màxims i mínims i freqüències en que apareixeran de tota la granja, per dia, per setmana, per mes i per any.

Aquest requeriment està associat a la fase 2 de les que es van presentar al capítol 1 (secció objectius).

3.2.3.- Selecció de les dades rellevants.

L'objectiu d'aquest requeriment és, entre totes el dades calculades a l'apartat anterior, cal escollir-ne les de més rellevància per a la simulació.

Aquest requeriment correspon a l'etapa 3 de les que es van presentar al capítol 1 (secció objectius).

3.2.4.- Introducció al simulador de les dades seleccionades.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

En aquest requeriment es voldrà crear una aplicació dins d'un bloc¹ de manera que automatitzi tota la feina anterior i introdueixi les dades seleccionades a l'apartat anterior als paràmetres d'entrada del simulador. Tot això es farà amb la incorporació de llibreries DLL creades en llenguatge C i incloses a la carpeta Extensions del simulador.

Aquest requeriment correspon a l'etapa 3 de les que es van presentar al capítol 1 (secció objectius).

3.2.5.- Desenvolupament de dos nous blocs: transició i engreix.

Aquest requeriment es correspon exactament amb la quarta etapa. El seu objectiu és la creació de dos nous blocs dins la llibreria: El de *Transició* i el d'*Engreix*.

El de *transició* serveix per simular el temps que hi ha entre el deslletat i el pas a l'engreix. Per la seva banda, el d'*engreix* serveix per simular el temps mentre es troba en l'etapa d'engreix. Es parteix del pes inicial i pes de sortida desitjat, de manera, que calcula el temps a transcórrer i deixa passar aquest temps fins que sortiran els garrins o porcs d'aquest bloc.

3.2.6.- Validació dels ensenyaments amb dades reals.

Aquest requeriment es correspon amb la cinquena etapa. EL seu objectiu és el de fer un procés de perfeccionament i millora o afinament de totes les etapes anteriors, de manera, que quedi de la forma més idònia, precisa i correcta possible. L'intenció és que s'assemblarà el més possible a la realitat i reflectirà també el valors i resultats correctes i reals en l'àmbit industrial.

¹Entenem per bloc un mòdul de l'aplicació que abstreu una fase de simulació.

3.3- Anàlisi de Requeriments NO Funcionals.

Els requeriments no funcionals, normalment són propietats o restriccions que ha de tenir l'aplicació, imposades pel client o pel mateix problema i que afecten al disseny. Normalment són quantificables i normalment garanteixen una major qualitat en l'aplicació. Aquestes propietats o requeriments no funcionals que trobem implícits a les operacions i funcions del sistema són:

- Consistència.** Tots els mecanismes que s'utilitzen, ho són sempre de la mateixa manera, per tant s'evita que l'usuari es confongui durant l'execució i realitzi un mal ús de l'eina.

- Disponibilitat.** L'aplicació estarà disponible per a ser utilitzada per qualsevol usuari. La restricció solament vindrà donada pel propietari de la mateixa aplicació i del projecte.

- Elasticitat en els canvis.** Tal i com ha passat amb el projecte anterior, s'ha desenvolupat i dissenyat l'eina actual utilitzant el disseny descendent i comentant totes i cada una de les línies codificades. D'aquesta manera en resultarà fàcil la lectura i la realització de futures modificacions o millores.

- Ergonomia.** L'aplicació és de fàcil instal·lació i maneig.

•**Fiabilitat.** L'aplicació garanteix que les dades generades són correctes i per tant els càlculs que se'n realitzaran també ho seran. A més, les rutines de càlcul han estat degudament comprovades després de la seva modificació.

•**Solidesa.** Els possibles errors que es produeixin durant l'execució són controlats pel sistema, el qual avisa a l'usuari i permet que aquest continuï amb la utilització de l'eina.

3.4 Implementació

Per poder començar hem d'establir un esquema o plantejament inicial del que volem fer.

Aplicacions en llenguatge C:

Les aplicacions en llenguatge C estaran enfocades en analitzar cadascuna de les etapes del cicle de la truja. Aquestes són la lactació, el detestat, l'engreix, la cubició, la gestació i el part. En cadascun d'ells s'analitzarà les dades llegides des d'un fitxer .VSI en funció de tota la granja, la setmana, el mes o l'any. En un annex es pot veure el format i l'estructura que té el fitxer .VSI. Entre les dades calculades seran la mitjana, la mediana, la desviació, el mínim, el màxim, les freqüències de cadascuna per un possible futur histograma.

De totes les dades ens fixarem en unes quantes que són les que ens importen i les utilitzarem pel simulador. Després s'introduiran al simulador llegint-les des d'un fitxer generat per un preprocessat i més tard, traslladar les dades d'aquí cap a els blocs o paràmetres que són les entrades del simulador. Un cop arribats a aquest punt, aquesta fase estaria acabada, comprovant, verificant i validant les dades obtingudes des de les funcions de càlcul.

Aplicacions en llenguatge Modl:

El llenguatge *Modl* és el llenguatge de programació orientada a objectes que utilitza el programa *ExtendSim* 6 similar la llenguatge de programació C. És un llenguatge que s'utilitza per crear blocs. Aquest llenguatge permet crear variables o connectors d'entrada i sortida, i també fer un tractament d'aquests valors. Un exemple d'aquest tractament és la programació del càlcul de l'equació Whitemore(1998,pp439).

Com a exemple de blocs que s'han creat comentarem les de la transició i de l'engreix. Per la banda del bloc de transició he volgut plasmar el temps que transcorre des de l'instant en que la truja deixa de detestar fins al moment en que la truja guanya prou pes(80 quilograms) per poder passar a l'etapa de l'engreix. En l'etapa de l'engreix transcorre els temps suficient fins que guanya els 100 o 120 quilograms). Després d'aquestes etapes la truja ja pot anar a l'escorxador. Els temps que transcorren en cada etapa depèn de les característiques de pes inicial, pes final i guany de pes. El llenguatge que s'utilitza per programar-los és el *Modl*.

Creació d'una DLL (Dyanmic Linking Libreriy)

Una DLL és una llibreria de software que es carrega a la memòria en temps d'execució, quan l'aplicació ho necessita.

En la creació d'una llibreria de tipus Dynamic Linking Library o DLL utilitzem el compilador de C de Windows DevC 4.9.9.2. Per crear-lo, primer de tot cal crear un projecte. Per això, es va a l'opció Archivo de la barra de menús, escollim Nuevo i en acabat Proyecto. Ens demanarà el nom del projecte i escollirem l'opció de crear una llibreria DLL en C. En acabat, demanarà el nom

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

del projecte amb extensió .dev. Després es crearà un fitxer de capçalera .h i un altre principal anomenat main.c. En acabat, si el compilem amb l'opció Ejecutar de la barra de menús i agafant l'opció Compilar ens crearà la nostra llibreria DLL si no hi ha errors de compilació. El contingut que hauria de tenir el fitxer de capçalera seria com aquest codi:

```
/* dll.h */

#ifndef _DLL_H_

#define _DLL_H_

#if BUILDING_DLL

# define DLLIMPORT __declspec (dllexport)

#else /* Not BUILDING_DLL */

# define DLLIMPORT __declspec (dllimport)

#endif /* Not BUILDING_DLL */

#endif /* _DLL_H_ */
```

En canvi, un exemple del fitxer main.h que retorna la suma de dos valors, seria el codi següent:

```
#include      "Windows.h"

/*****

/* This is an exported routine for the DLL.

*/

*****/

__declspec(dllexport) double add(double x, double y)

{

return(x+y);

}
```

La generació de llibreries DLL ha plantejat dificultats en la realització del bloc d'entrada de dades. Això és perquè a l'hora de crear la llibreria DLL formada pel fitxer de capçalera i el principal a través del programa DevC, no es podia executar o compilar degut a que el sistema operatiu Windows Vista no ho permetia . La solució ha estat compilar-lo i executar-lo desde el sistema operatiu Xp.

Fitxer de dades de la granja.

Fitxer Vsi des d'on s'ha extret les dades per extreure'n les dades per tal de realitzar els càlculs estadístics que s'han tingut en compte per introduir les dades a l'aplicació. El format del fitxer es troba a l'annex II.

3.5. Dificultats en la implementació

Durant la implementació de cada etapa del projecte s'han produït algunes dificultats que se sintetitzen següentment:

- 1) La falta d'informació inicial. Estudi per obtenir tots els coneixements necessaris per saber el tema que es vol tractar. Per exemple, veure els llibres de la bibliografia i fer-se una idea del que es vol simular.
- 2) Problemes en la consecució de les estadístiques generades. Temps de dedicat a llegir el fitxer en vsi i treure'n les dades correctes , comprovades en excel , verificades i validades, i fer-ne els càlculs i treure'n estadístiques adients.
- 3) Mal funcionament de la llibreria dinàmica DLL. Part que més temps ha portat a terme, no només la dificultat tècnica sinó mes aviat per que el sistema operatiu Windows-Vista donava dificultats a l'hora de crear fitxers DLL i executar-los en el simulador. També s'ha comprovat que aquesta dificultat no existia en el cas del sistema operatiu Xp.
- 4) Falta de realitat de les dades simulades. Verificar i validar les dades amb la realitat que es pretén simular.

3.6- Recursos del Sistema.

3.6.1.- Recursos Hardware.

Per a desenvolupar aquesta aplicació s'han utilitzat una màquina, proveïda del software necessari.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Les característiques i els recursos de l'ordinador principal, què és el que més cops s'ha utilitzat en el desenvolupament i que permeten la utilització de l'eina sense problemes són:

Característiques ordinador principalProcessador Intel Pentium II - 2 GHz.Monitor color Súper VGA de 15'.Ratolí i teclat estàndard.Disc Dur principal de 4 GB.Disc Dur secundari de 320 GB.2 GB de memòria RAM.

Un cop s'ha descrit l'ordinador utilitzat per al desenvolupament de l'eina, cal dir quin seria l'equip mínim per tal de que l'aplicació pogués funcionar. Això significa que sota un equip de les següents característiques, es podria utilitzar el programa, però aquest funcionaria amb gran lentitud, cosa que reduiria la seva eficiència.

Característiques de l'equip mínimProcessador Pentium II. Monitor color Súper VGA de 15'.Ratolí i teclat estàndard.Disc Dur de 80 GB.1 GB de memòria RAM.

3.6.2.- Recursos Software.

L'aplicació desenvolupada treballa sobre entorn Microsoft Windows Vista o superiors, havent-se alternat també en la seva creació Windows XP, sobretot , i Windows 98.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

L'eina amb la que s'ha basat el desenvolupament ha estat el *ExtendSim* 6. En el nostre cas, l'hem triat, ja que com que aquest projecte es tracta de la creació d'un simulador d'una assignatura de segon cicle ja existent.

La taula següent conté les aplicacions que s'han usat per desenvolupar l'eina.

Programari utilitzat en el desenvolupamentSistema Operatiu Vista i Xp .Dev C++
4.9.9.2.*ExtendSim* 6

A part, els programes utilitzats per al desenvolupament d'aquesta documentació, així com per l'exposició del treball de final de carrera són:

Programari utilitzat en el desenvolupamentMicrosoft Word 2007.Microsoft Excel
2007.PaintBrush.*ExtendSim* 6

3.6.3.- Restriccions tècniques.

Hi ha un conjunt de restriccions tècniques que pot sofrir el software. Aquestes restriccions poden venir donades sense que el disseny ni la codificació en tinguin res a veure. L'aplicació actual també pot sofrir aquestes restriccions, que són:

•**Velocitat de resposta.** L'eina realitza tots els càlculs mitjançant càlculs de C i de modL bastant complexes. S'ha provat els temps que transcorre es de 0,5 segons.

•**Memòria.** L'aplicació no consumeix molts recursos de memòria. Com hem dit en els recursos hardware, 1 GB de memòria RAM (vegeu característiques de l'equip mínim) són suficients pel correcte funcionament de l'aplicació, per tant no existeix problema de memòria, tal com s'ha comprovat. El càlcul de la quantitat de memòria RAM utilitzada per l'aplicació es dificulta pel fet que aquesta aplicació usa llibreries de càlcul i gràfiques que han estat implementades per tercers i de les quals desconexim els seus requeriments de memòria. En l'administrador de tasques indica que ocupa 14,8 Megabytes.

4.- PLANIFICACIÓ DEL PROJECTE.

L'objectiu de la planificació del projecte de software és proporcionar un marc de treball que permeti al gestor fer estimacions raonables de recursos, costs i planificació temporal.

Aquestes estimacions s'haurien de fer dins d'un marc de temps limitat al principi d'un projecte de software i s'han d'anar actualitzant regularment a mesura que aquest progressa.

L'objectiu de la planificació s'aconsegueix mitjançant un procés de descobriment de la informació que porti a estimacions raonables.

4.1.- Identificació i Gestió de Riscs.

En tot projecte de desenvolupament de software, existeixen riscos que poden fer fracassar aquest projecte. En els següents apartats tractarem de definir quins són aquests riscos, com ens afecten i que podem fer per a que això no succeeixi.

4.1.1.- Identificació dels riscos.

La identificació d'un risc consisteix en enumerar els riscos concrets d'un projecte. Així doncs, analitzarem si en el nostre projecte podem trobar els riscos que a continuació descriurem.

- No poder complir la data de lliurament de l'aplicació i la seva corresponent documentació.
- Disminució del número de persones a l'equip de desenvolupadors.
- Poc rendiment d'algun component de l'equip de desenvolupadors (falta de temps, treball excessiu, tasca massa gran per a una sola persona, ...).
- Canvis en les especificacions motivats per l'aparició de noves necessitats durant la codificació.
- Utilització d'una nova eina de desenvolupament no utilitzada mai abans pel desenvolupador, i que requereix un llarg període d'aprenentatge durant el transcurs de la codificació.
- Incompatibilitats o funcionament anòmal de l'aplicació al passar del hardware/software del desenvolupador al hardware/software de l'usuari final.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

- Aplicació per primer cop d'un nou mètode de control de qualitat (normativa ISO 9000-3). Aquest, com que és molt més rigorós que els mètodes utilitzats fins al moment, poden incrementar l'esforç per part del tècnic, a més d'analitzar les necessitats del client i plasmar-les en la documentació prevista en la fase de codificació.

- Canvi o ampliació dels requisits per part del client. Es podria donar el cas que un cop identificats els requisits i especificat la implantació corresponents a aquests, el client decidís modificar els requisits o fins i tot afegir-ne més. Encara que aquest risc és mínim en les primeres fases del desenvolupament, com més avança el projecte, més important serà l'impacte sobre el mateix.

- Problemes de certa importància en la codificació. Els problemes de la fase de codificació, que ja estan en principi previstos a la planificació del projecte, es poden veure augmentats per diversos factors.

- Client – Desenvolupador. És el primer cop que el client (en aquest cas, el director del treball) i el desenvolupador treballen junts. El desconeixement inicial del cicle de vida de centres de producció de garrins per part del desenvolupador, comporta el risc de que no s'entenguin completament els requeriments ni el funcionament de l'aplicació. Això representaria un esforç i temps de formació major.

Tot això pot influir a la data de lliurament i per tant, produir un cost afegit associat que incrementarà el cost previst.

4.1.2.- Projecció dels riscos.

En la següent taula, es pot observar la probabilitat de que un risc es pugui produir, així com la seva importància.

Pot donar-se el cas de que algun risc no s'hagi tingut en compte.

Figura 5: Taula de riscos i ponderacions.

La ponderació significa la importància del risc en cas de que aquest es donés. En el nostre cas, com més problemàtic és el risc, major ponderació. Això significa que primer hem de tractar d'evitar riscos amb major ponderació, i així successivament.

4.1.3.- Gestió de riscos.

En el cas de que es produís algun dels riscos identificats anteriorment, la metodologia que es seguiria és:

- Incompliment de la data de lliurament.* En el nostre cas, ens trobem davant d'un treball de final de carrera d'un alumne de l'EUP. Tot i que l'eina que es desenvolupa serà utilitzada per el Departament de Matemàtica, Grup de model de Decisions, Computació i Simulació, i es desitja tenir-la amb disponibilitat el més aviat possible, el principal interessat en acabar el projecte a temps és l'alumne. Tot i això es parlarà amb el client per tal d'ampliar el termini de lliurament.

- Poc rendiment del desenvolupador.* En el nostre cas, el baix rendiment de l'únic desenvolupador per causes com la realització de feines paral·leles o la falta de temps a l'hora de desenvolupar el projecte, produiria un retard en el desenvolupament de l'eina i per tant un retard de la data d'exposició del treball de final de carrera. Aquesta raó és suficient motivació com per a controlar el temps de desenvolupament.

- Canvis d'especificacions.* Un cop detectada la nova necessitat durant la modificació, s'estudiarà la situació i es buscarà una solució. Si cal, es realitzarà un annex a la documentació de les especificacions i a partir d'aquest annex es podrà seguir el curs normal de la codificació.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

- Nova eina de desenvolupament.* En el cas de que es tingui dificultats en l'adaptació a la nova eina de desenvolupament, s'optarà per dedicar un temps addicional i no planificat per a l'aprenentatge. Aquest aprenentatge pot consistir en consultar bibliografia, manuals, utilització de l'ajuda de l'eina i/o sol·licitar ajuda a algun altre desenvolupador més experimentat i avançat.
- Mal funcionament de l'aplicació.* Si l'aplicació no funciona correctament al hardware/software del client, es dedicarà un temps extra en detectar i solucionar el problema.
- Nou mètode de control de qualitat.* Si es dóna aquest problema, l'única solució viable és, a l'igual que en el cas d'una nova eina de desenvolupament, dedicar un temps addicional a l'enteniment i aprenentatge del mètode.
- Canvi o ampliació de requisits.* Si es produeix aquest cas, immediatament es realitzarà una reunió amb el client per analitzar-ho. Posteriorment s'estudiarà la situació per part del desenvolupador, el qual realitzarà un annex a la documentació corresponent.
- Problemes de codificació.* Obtenir una visió més general de la tasca i a partir d'aquest punt, fer un replantejament de l'estructura dels modes de l'aplicació, per poder localitzar i puntualitzar més ràpidament la causa del problema.
- Nou client.* Es buscarà un model alternatiu per l'ajuda de l'aplicació i del manual d'usuari, així com un element de comunicació permanent entre el client i el desenvolupador.

4.2.- Planificació.

A part d'analitzar amb tot detall els possibles riscos del projecte, també cal analitzar el treball a realitzar per poder desenvolupar correctament l'aplicació actual.

Per analitzar el treball, calcularem el temps estimat per a la seva realització, dividint el treball en diferents fases, i indicant la duració de cadascuna d'aquestes.

Aquesta estimació és aproximada, tot i que durant la confecció del projecte es van variant els resultats fins aconseguir mesures més exactes.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Figura 6: Taula de tasques del projecte.

5.- PROVES DE VALIDACIÓ DEL SISTEMA.

Una de les parts més importants en el desenvolupament d'un producte software és la validació del sistema. Aquesta validació del sistema es realitza mitjançant Proves.

La Prova és un element crític per a la garantia de qualitat del software i representa una revisió final de les fases d'especificació, anàlisi i disseny. Així doncs, la Prova ens serveix per avaluar el que ja tenim fet, i la seva importància radica en l'alt cost de correcció d'errors. El que s'intenta en les proves és fer fracassar el que hem construït i demostrar que hi existeixen errors.

D'aquesta manera podem concloure que els objectius de la Prova són:

- Detecció dels errors.
- Disseny dels casos de prova per trobar errors en el mínim temps i esforç.
- Validar les especificacions i el rendiment de l'aplicació.

Ens podem trobar diferents tipus d'errors després de la fase d'avaluació:

- Errors greus. Són deguts al disseny. La viabilitat i qualitat serà qüestionada. Això implicarà la realització de noves proves.
- Errors fàcilment corregibles. Ens trobem en que la viabilitat i qualitat seran acceptables o les proves són inacceptables per trobar errors greus.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

- No errors. Això implica que no hi ha hagut un bon disseny dels casos de prova.

Per realitzar una validació és important seguir una estratègia. Aquesta ha d'estar planejada en tot moment per tal d'aconseguir un bon sistema de prova i trobar el màxim d'errors possibles.

Els tipus de proves que formen part d'una estratègia de prova són:

- Proves de Baix nivell: Són les que verifiquen cada segment de codi.
- Proves de Alt nivell: Són les que validen els requeriments funcionals.

L'estratègia de prova que s'ha utilitzat està format per les proves:

- Prova d'Unitat.
- Prova d'Integració.
- Prova Validació.
- Prova de Sistema.

Les dues primeres són les anomenades proves de baix nivell, mentre que les dues darreres formen part de les d'alt nivell.

5.1.- Prova d'Unitat.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

S'aplica a cada mòdul del programa. S'utilitza en els principals camins de control i s'utilitza les anomenades proves de caixa blanca. Això significa que la prova es centra en l'estructura de control del programa, totes i cadascuna de les sentències i els mòduls del programa.

Així doncs es proven les interfícies per tal de:

- Verificar que la informació flueixi de forma adequada cap a i des del mòdul que estem provant.
- Verificar les estructures de dades locals per tal de garantir que les dades es mantenen temporalment conservant la seva integritat.
- Trobar diferents camins independents i obtenir així errors del tipus de comparacions entre tipus de dades diferents, operadors lògics, ...
- Verificar les condicions límits en cas de bucles.

En el projecte hem dissenyat casos de prova que ens han garantit que s'ha executat com a mínim un cop cada camí d'un mòdul i les condicions lògiques en les seves vessants vertaderes i falses i els bucles en les condicions límits.

S'han provat les interfícies per a que les dades i les crides flueixen de manera correcta i la interacció de les dades amb les bases de dades.

5.2.- Prova d'Integració.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Un cop s'han provat tots els mòduls individualment, aquests s'han d'unir per formar el programa. Un cop fet això, s'han de provar si funcionen correctament. L'objectiu, doncs, és construir l'estructura del programa.

En el sistema s'han integrat els mòduls de forma incremental i descendentment en primer en profunditat. D'aquesta manera, es construeix i es prova l'aplicació en petits segments on els errors són més fàcils d'aïllar i corregir. Això permet plantejar una prova sistemàtica.

S'ha començat pel mòdul principal (el que engloba tots els requeriments funcionals i el menú del sistema) i s'han anat incorporant mòduls en primer en profunditat; en cada mòdul integrat s'han anat fent proves per tal de verificar el seu bon funcionament i la seva interacció amb els altres mòduls integrats.

5.3.- Prova de Validació.

A partir d'aquest tipus de prova, s'utilitzen tècniques de prova de caixa negra. Això significa que s'avalua tot el programa en conjunt, validant els requeriments funcionals i provant l'àmbit d'informació de tot el programa.

La validació s'aconsegueix quan el software funciona tenint en compte les expectatives dels clients, les quals estan definides en l'especificació dels requeriments.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Es pretén provar són els requeriments funcionals, els requeriments de rendiment de tota l'aplicació, que la documentació sigui correcta, que sigui intel·ligible i a més provar la portabilitat, facilitat de manteniment, ...

Aquí es porten a terme les proves Alfa i Beta. En la prova Alfa el client utilitza el programa en el lloc de desenvolupament de forma natural i el desenvolupador anota els errors que es van produint.

La Prova Beta es dur a terme al lloc del client i és el client qui registra els errors.

En aquest cas, el client ha revisat les proves, notificant tot seguit els errors trobats al programador.

5.4.- Prova de Sistema.

El software és integrat en altres elements com el hardware, informació, ... i es realitzen una sèrie de proves per verificar la integració a tot el sistema.

Dins de les proves de sistema tenim:

- Prova de Recuperació. En cas de fallida ens diu com es recupera el sistema.
- Prova de Seguretat. Comprova introduccions no permeses.
- Prova de Resistència. Es porta el programa a casos extrems.

- Prova de Rendiment. Per aplicacions en temps real.

5.4.1.- Prova de Recuperació.

La prova de recuperació força l'error del programa i verifica que la recuperació es porti a terme adequadament ja sigui de forma automàtica, del mateix sistema, o manual mitjançant l'usuari.

En cas de que es produís una caiguda del sistema, l'aplicació es tancaria i el programa vetllaria per la integritat de les dades.

La prova que s'ha realitzat és la de donar robustesa al sistema utilitzant sentències de l'estil si hi ha error digues-ho sinó executat.

5.4.2- Prova de Seguretat.

La prova de seguretat intenta verificar que els mecanismes de protecció incorporats al sistema s'encarreguin de les penetracions impròpies.

Els requeriments funcionals d'aquesta aplicació no demanaven cap autenticació d'usuaris. Per tant, no s'ha fet una prova de seguretat.

5.4.3- Prova de Resistència.

Les proves de resistència estan dissenyades per enfrontar el sistema a situacions anormals o extremes i veure com aquest respon. S'executen de manera que demanin recursos en quantitat, freqüència o volums anormals.

Aquesta prova no s'ha dut a terme.

5.4.4- Prova de rendiment.

Aquest tipus de prova s'utilitza en sistemes basats en temps real. En les proves anteriors hem provat els mòduls per separat, però fins que no ho tenim tot unit, no podem fer una prova final.

La prova de rendiment està dissenyada per provar el rendiment del producte en temps d'execució dins del context d'un sistema integrat.

Es va veure que el programa responia correctament en qualsevol sistema integrat . En alguns anava més ràpid i en altres més lents però això depèn del processador i la memòria del sistema i no del propi programa.

6.- GARANTIA DE QUALITAT DEL SOFTWARE.

L'objectiu de l'enginyeria del software és obtenir un software de qualitat. Podem definir-la com SQA (Software Quality Assurance). D'obtenir un software de qualitat se n'encarrega la garantia de qualitat del software, que pretén ser una sèrie d'activitats per garantir que s'obtindrà un producte de qualitat.

El que engloba la garantia de qualitat del software és:

- Mètodes i eines de disseny i anàlisi.
- Revisions tècniques formals.
- Estratègies de prova.
- Control de la documentació.
- Ajust a estàndards.
- Mecanismes de mesura.

Quant parlem de la Qualitat del software, parlem de requeriments explícits, estàndards i requeriments implícits. Els requeriments explícits venen marcats al inici del projecte. Són els requeriments inicials del client. Els estàndards són una sèrie de mesures per a que el nostre software s'hi adapti, i els requeriments implícits no venen marcats en lloc, però s'han de complir, com el manteniment del nostre producte.

Els factors que determinen la qualitat del software, és poden classificar en:

- Característiques Operatives.
- Capacitat d' Adaptació a d'altres Entorns.
- Capacitat d' Adaptació a Canvis.

6.1.- Característiques Operatives.

Les característiques operatives o factors que determinen la qualitat del nostre software que venen marcats pel grau o nivell de les següents característiques, són:

•**Correcció:** Permet mesurar si el programa satisfà els requeriments inicials i objectius del client.

La realització dels testos que hem presentat al capítol anterior assegura un nivell adequat d'assoliment d'aquesta característica.

•**Viabilitat:** En aquest cas, es mesura el grau que una aplicació realitza les seves funcions amb la precisió requerida.

La prova de validació indicada al capítol anterior, en què el client va comprovar els resultats de la simulació obtinguda per l'aplicació mostren un nivell adient de viabilitat de l'aplicació.

•**Eficiència:** Quantitat de recursos d'ordinador i codi requerits per a que el programa realitzi les seves funcions.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Els resultats són obtinguts per l'aplicació de manera quasi immediata, fins i tot en els sistemes més lents amb què s'ha provat.

•**Integritat:** En permet valorar el grau en que es pot controlar l'accés al software i a les dades per persones no autoritzades.

Dins dels requeriments de l'aplicació no hi havia l'autenticació dels usuaris. Per tant, la característica d'integritat no té tant de sentit en aquest cas.

•**Facilitat d'ús:** Serà l'esforç necessari per aprendre a utilitzar el programa i a entendre'n les sortides.

L'ús de l'aplicació és força intuïtiu ja que es basa en la utilització de blocs preexistents connectats entre ells de forma seqüencial seguint les etapes de la vida del porc. La introducció de dades a cadascun dels blocs és senzilla i dirigida per la pròpia aplicació. La introducció de les dades inicials tampoc planteja cap dificultat. A més a més, disposa d'un manual d'usuari.

6.2.- Capacitat d'Adaptació a d'altres Entorns.

Els factors que influeixen en la qualitat del producte creat en aquest treball final de carrera són:

•**Portabilitat:** L'esforç necessari per portar el programa a d'altres màquines que tingui instal·lat el sistema operatiu Windows és mínim, només cal instal·lar l'*Extend* al nou

ordenador on es vulgui instal·lar-lo. Aquesta instal·lació és molt senzilla, el manual de l'*Extend* indica com fer-la.

•**Reusabilitat:** El disseny modular de l'aplicació en forma de blocs fa que resulti senzilla la reutilització d'aquests blocs per ampliar l'aplicació. De fet, per desenvolupar-la s'han reutilitzat blocs preexistents d'una altra aplicació anterior.

•**Interoperativitat:** L'esforç per afegir el nostre sistema a d'altres sistemes és petit ja que el llenguatge de programació en que s'ha desenvolupat aquest producte (Borland C) és un llenguatge que s'adapta a d'altres entorns i Sistemes Operatius com els Sistemes Operatius UNIX (Linux, Minix..).

6.3.- Capacitat d'Adaptació a canvis.

Els factors que ens determinen si el software desenvolupat és fàcil o difícil de mantenir són:

•**Facilitat de manteniment :** La detecció i correcció d'errors ha tingut un baix cost ja que gràcies a les revisions en que s'ha sotmès el software desenvolupat en les etapes de disseny i anàlisi fan disminuir el cost final.

•**Flexibilitat:** El software desenvolupat és flexible a la incorporació de canvis. La programació modular del programa i la estructuració en les dades fan que sigui un programa flexible.

- Facilitat de prova:** Les proves de validació, abans esmentades, han tingut un cost baix en l'esforç del producte final.

7.- CONCLUSIONS

La finalitat del treball final de carrera ha estat la de crear un simulador amb el que es poden obtenir els resultats més pròxims als que s'assolirien en la realitat si, en comptes d'utilitzar el simulador, es fes amb proves reals.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Per aquesta raó els ordinadors son una eina molt útil. Per mitjà d'ells podem assolir decisions de com voldríem que funcionés la nostra granja de porcs sense experimentar amb ells.

D'aquesta manera, amb la senzillesa del treball amb aquest simulador, es poden prendre decisions de forma virtual i comprovar-ne els resultats sense haver de comprometre inversions o haver d'esperar el pas del temps. L'ús del simulador esdevé així, molt important perquè d'una presa de decisions correcta en depenen els resultats que s'obtindran a la granja de porcs.

Un cop s'han entrat totes les fórmules i càlculs del model matemàtic de tot el sistema; un cop s'han creat cadascun dels blocs en què es basa el simulador i s'han connectat tots ells de manera coherent es pot tenir la garantia de que els resultats del simulador reflecteixen el que ocorrerà a la realitat. Com més detallat és el sistema i verificat, més fiable serà.

D'aquesta manera, amb aquest simulador, podem prendre decisions per verificar quina podria ésser l'opció que aportaria més grans beneficis a partir d'uns requeriments i condicions inicials. De l'estudi podem extreure conclusions del comportament del sistema que volem crear.

El simulador que s'ha desenvolupat en aquest treball final de carrera consta de cinc fases . Cadascuna d'aquestes fases s'ha fet el següent:

- Recollida de requeriments.* En aquesta fase es produeix una primera presa de contacte entre l'alumne desenvolupador i el professor. El que es pretén és que l'alumne es faci una primera idea dels requeriments que l'eina a desenvolupar necessita.
- Anàlisi.* En aquesta fase s'estudien els requeriments anteriors per tal de comprendre amb tot detall quina ha de ser la funcionalitat de l'eina.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

- Disseny.* A partir d'aquesta fase es prepara el disseny de l'eina. D'aquesta manera s'aconsegueix major facilitat a l'hora d'implementar i s'obté una més gran flexibilitat si ca, en un futur, realitzar canvis per causes no previstes.

- Implementació.* En aquesta fase es tradueix a codi C i *Extend* tot el disseny realitzat en la fase anterior. Depenent del disseny, aquesta fase serà més o menys fàcil i ràpida d'implementar.

- Proves.* Si partim de la base de que tota aplicació desenvolupada conté errors, aquesta fase ens permetrà localitzar aquests errors i solucionar-los. La utilitat final és la de localitzar i solucionar el major nombre d'errors possibles, ja que fer-ho en la seva totalitat és impossible.

El resultat és un simulador de cicle tancat de la producció porcina. En ell es detalla i comprova el cicle de vida d'un porc desde que neix fins que mor a l'escorxador, per una banda, i per l'altra la seva cria i reproducció continua.

Com a treball futur destacaria els elements següents:

Millora del producte. Aquest simulador es una bona eina per aconseguir una millora en la producció del porc, millorant l'alimentació del porc i per tant, millorant el producte final.

Millora dels processos de producció. En aquest simulador es pot modificar i prendre decisions sobre els processos de la producció. De manera, que es podem modificar i millorar per tal d'obtenir més beneficis i abaratint els costos.

Predicció de resultats. En aquest simulador podem veure els resultats que obtindrem del nostre sistema de producció i presa de decisions sense tenir que construir el sistema productiu en realitat, és a dir, podem comprovar els resultats de forma virtual.

Predicció de beneficis. Amb aquest simulador podem veure quins resultats obtindrem

després de prendre les nostres decisions i de crear el nostre futur sistema de producció de porcs, i quins beneficis n'obtindrem.

8.- BIBLIOGRAFIA

SIMULATION MODELING AND ANALYSIS, Averill M. Law i W. David Kelton, 2000, McGraw-Hill.

LA INFORMÀTICA A LA GRANJA DE PORCS, Roger Torres i Graell , 1985. Dilagro.

MANUAL IMPRESCINDIBLE C/C++, Miguel Ángel Acera García i Ana María Sanz Sierra, 2005, Anaya.

PRODUCCIÓN DEL CERDO PRINCIPIOS CIENTÍFICOS Y PRÁCTICOS, Colin T. Whitemore, 1998, Aedos.

CIENCIA Y PRÁCTICA DE LA PRODUCCIÓN PORCINA, Colin Whitemore, 1996, Acribia.

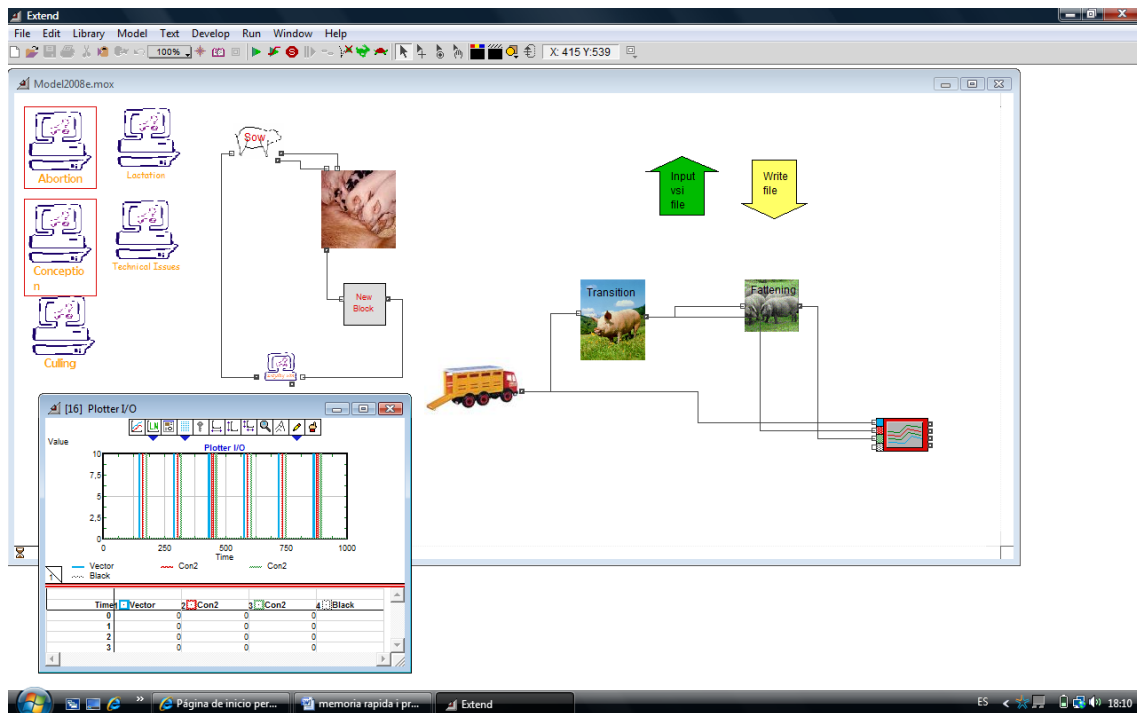
REPRODUCCIÓN DEL CERDO, P. E. Hughes i M.A. Varley, 1984, Acribia.

ANNEX I: Visualització de l'eina o de l'aplicació implementat.

Aplicació general:

En aquest aplicació he implementat la simulació d'un procés d'engreix d'una truja o d'un conjunt de truges. Es parteix d'uns paràmetres introduïts pel bloc Input file vsi que introdueix les dades d'un fitxer d'extensió vsi o Excel. A partir de llavors s'executa la simulació i es veuen en la gràfica els resultats. Finalment les dades importants son imprimides en un fitxer a través del bloc Write file.

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.



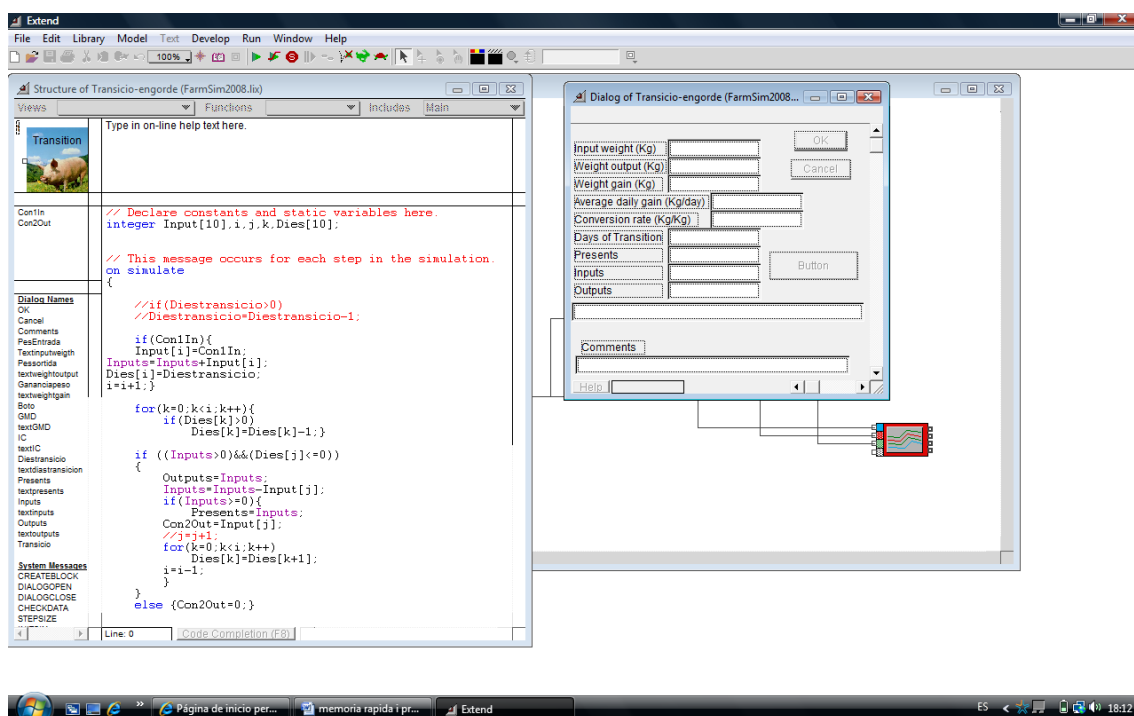
En la gràfica dels resultat veiem el temps en que les truges surten del procés de desllet i son introduïdes al bloc de transició. Després segons les característiques tècniques surten d'aquest bloc passat un temps i son introduïdes al següent bloc d'engreix. Passat un temps surten del bloc i van a parar a l'escorxador. Els resultats es veuen a la gràfica en funció del temps.

Interior de cada bloc:

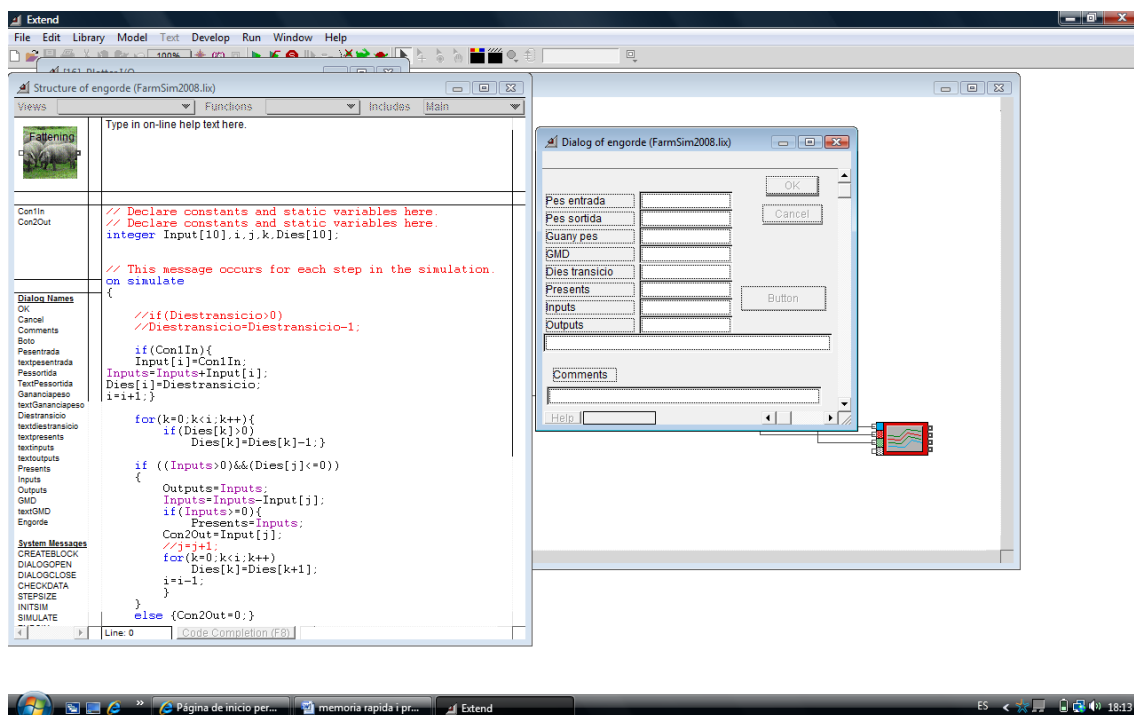
D'ara endavant ensenyaré l'interior de cadascun dels blocs que formen part del procés o de l'aplicació. Esta presentat de forma gràfica cadascun dels blocs per tenir una idea de com han estat creats. Els uns han estat creats totalment i d'altres han tingut algunes modificacions i millores. En l'etapa de perfeccionament i afinament l'he acabat de millorar-ho en la mesura del temps i del possible

Bloc de Transició:

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

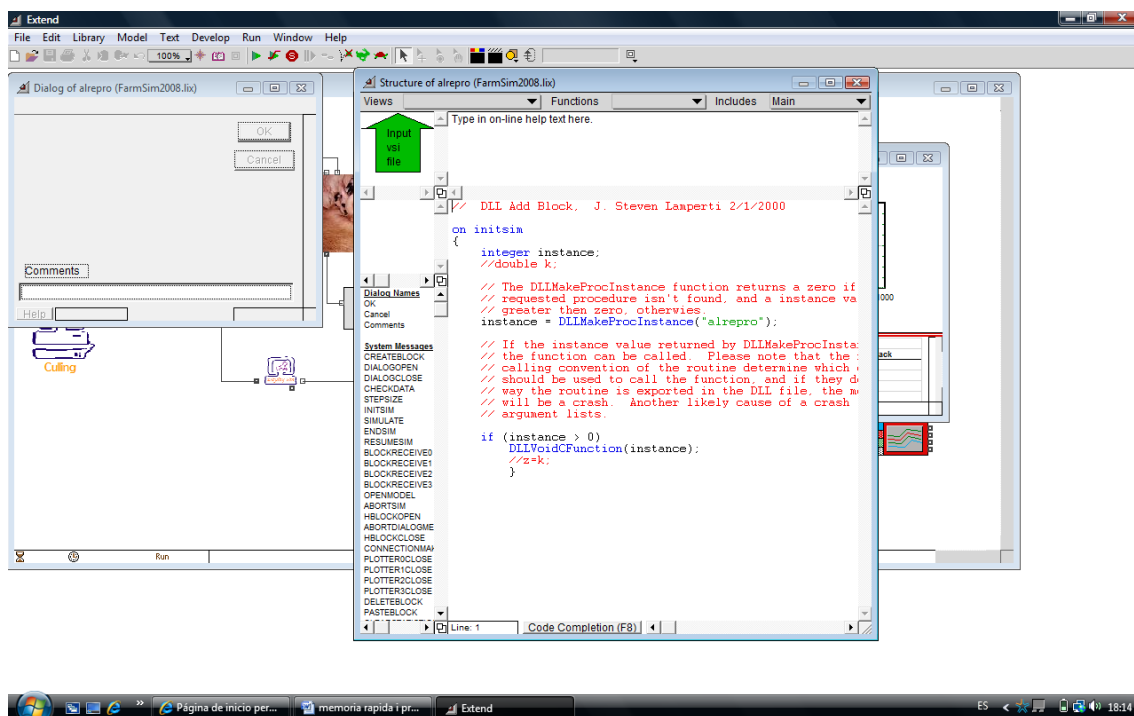


Bloc d'engreix:



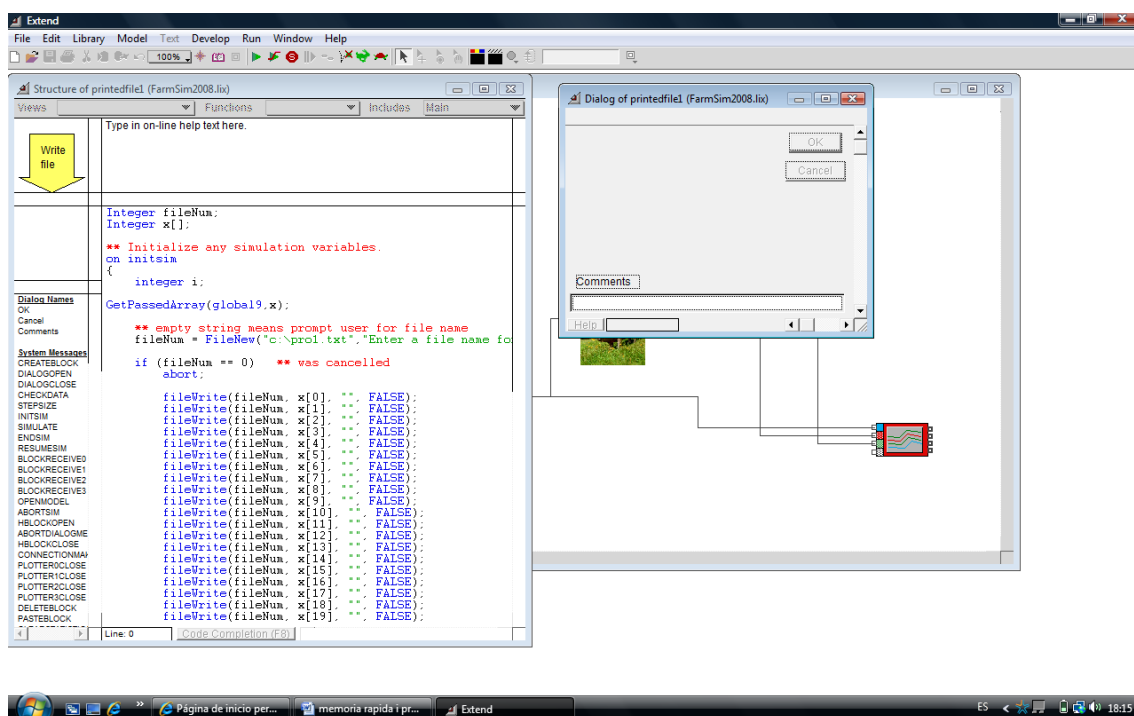
Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Bloc de Input:

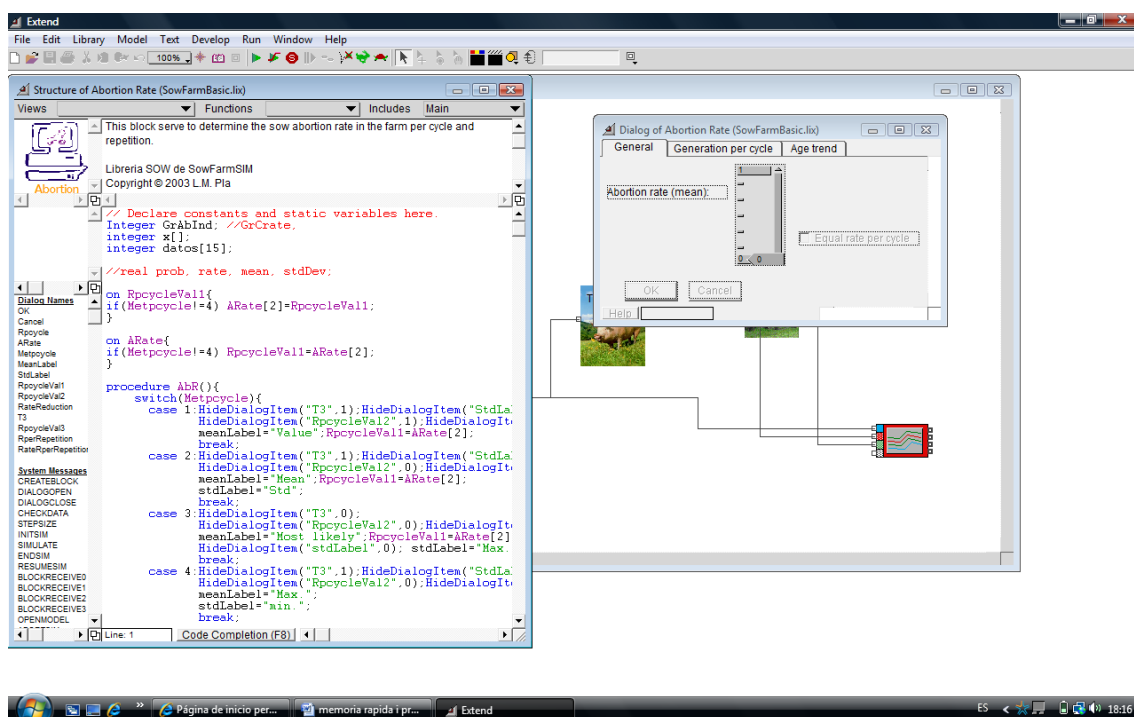


Bloc d'output:

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

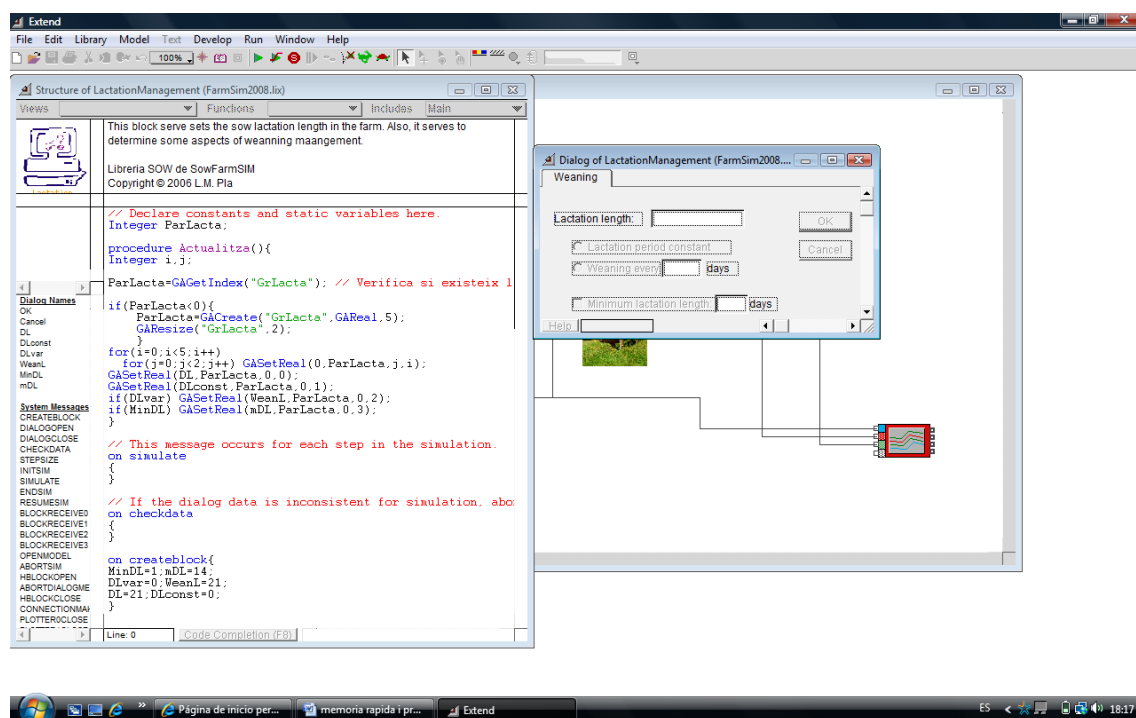


Bloc d'abort:

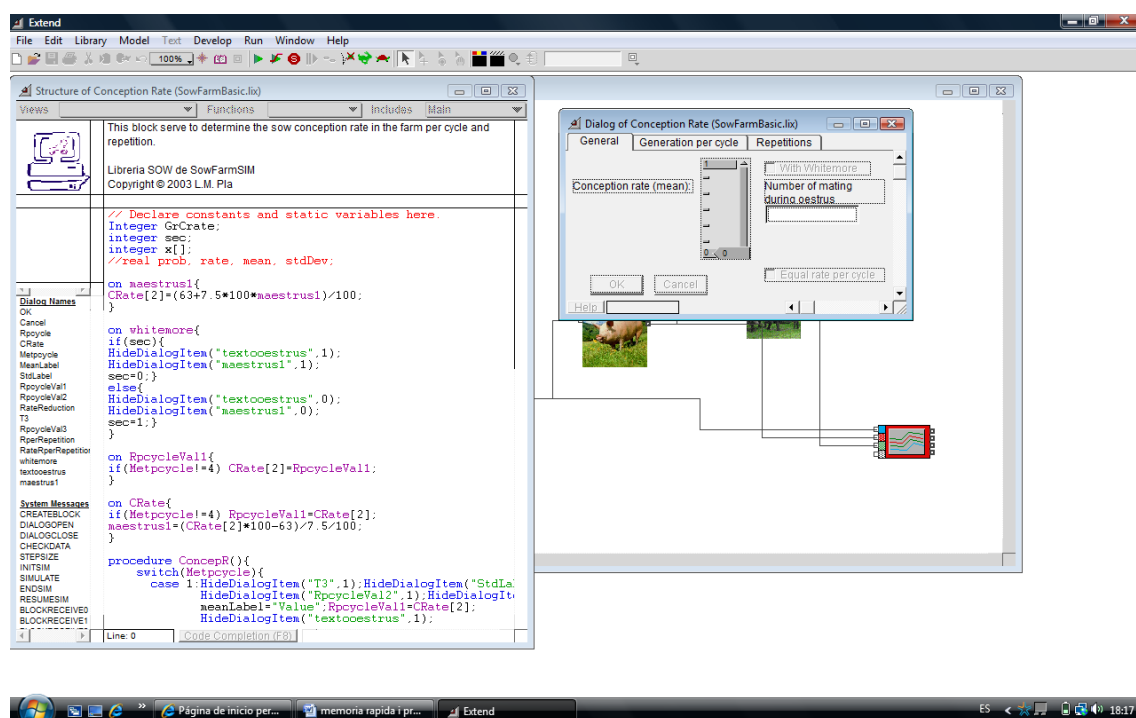


Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Bloc de lactació:

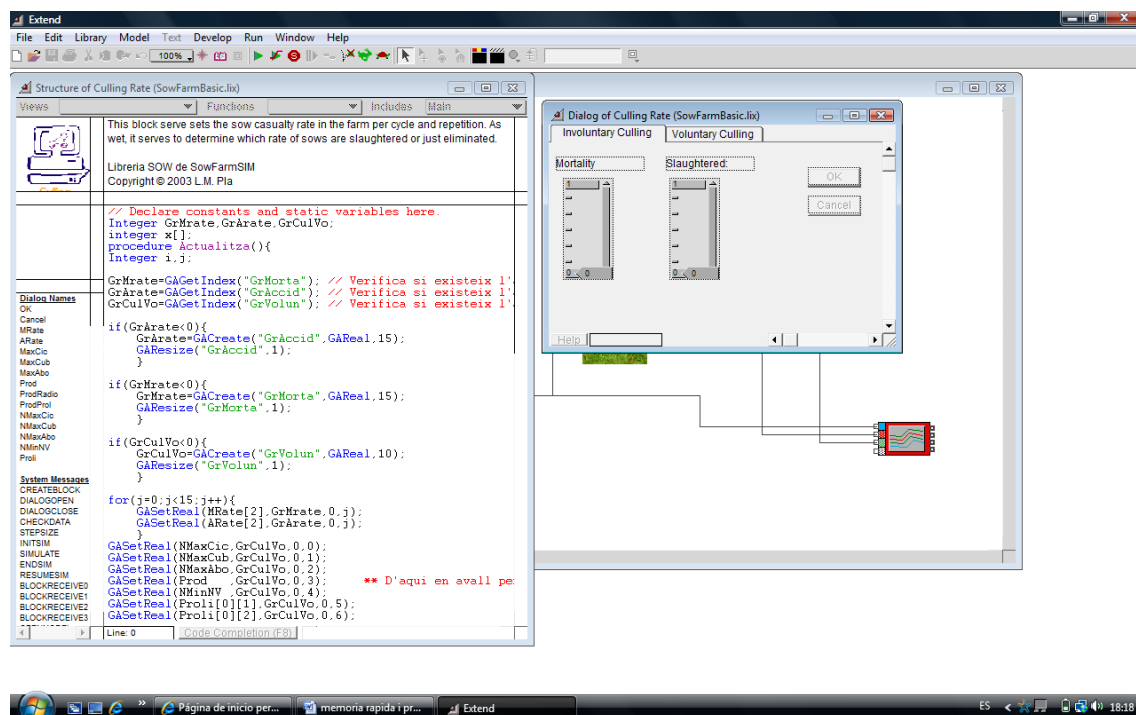


Bloc de concepció:



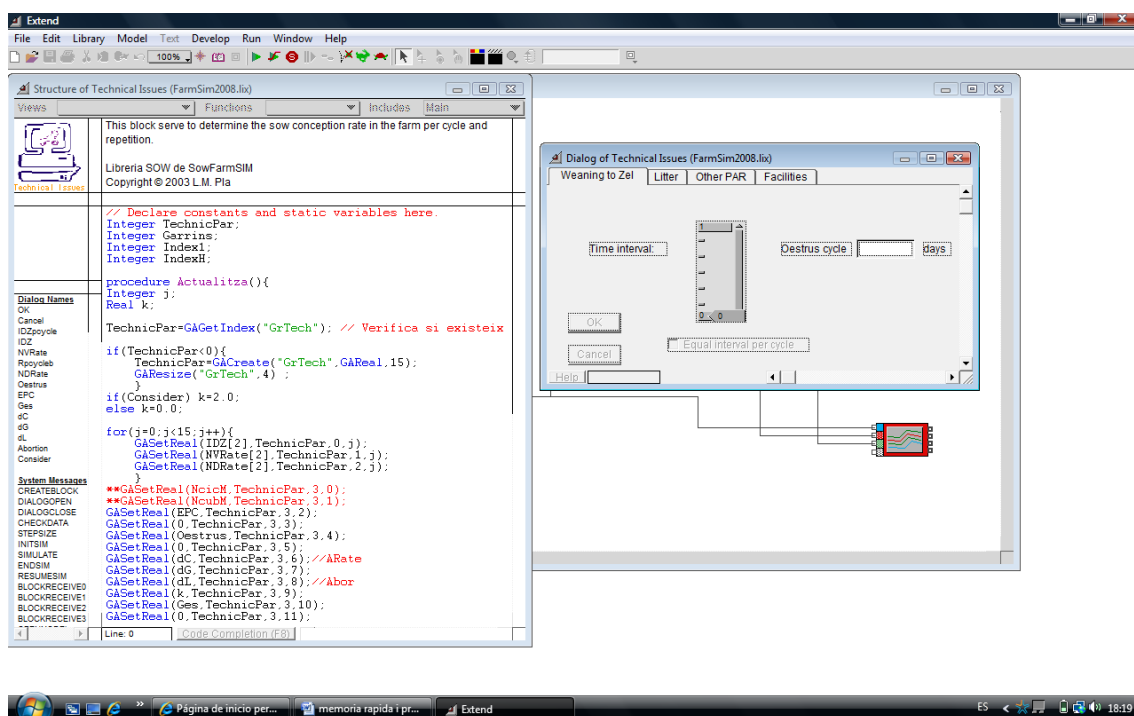
Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

Bloc de culling:



Bloc de technical Issues:

Estudi i implementació d'un simulador de cicle tancat de producció de porcs.



ANNEX II: FORMAT DEL VECTOR ESTANDARD D'INFORMACIÓ TÈCNICA – V.S.I.–

ALTAS

Format VSI lineal

0	5	11	13	14	16	22	27	33	34
40	42	128							

GRANJA: *Nº de Granja.* És un camp de 5 dígit que ens indica el nombre de granja

ANIMAL: *Nº de Animal* Es un camp de 6 dígit, distribuïts en,

Año: Última xifra de l'any de la data de naixement (1 dígit)

Orden: En el cas de que un nombre d'animal es repetís el nombre de ordre seria diferent, sinó por defecte se posaria el 0 (1 dígit)

Nº del animal : Nombre d' identificació del reproductor. (4 dígit)

CICLO: *Nº de ciclo*. Ens indica el nombre de cicle en el que es troba l'animal en el moment de la alta, normalment es 00. (2 dígit)

SEXO: *Sexo del Animal*. Es un nombre identificador del sexe del animal. (1 dígit)

1 hembra.

2 Macho.

RAZA: *Raza del Animal*. Camp de 2 dígit numèrics que indica la raça del **pare** y la **mare** del reproductor en aquest ordre.

1 = Large White.

2 = Landrace "materno".

3 = Landrace Alemán.

4 = Landrace Belga.

5 = Duroc.

6 = Pietrain.

7 = Hampshire.

8 = (altra raça).

9 = animal cruzado.

FECN: *Fecha de Nacimiento*. Camp de 6 dígitos numèrics para la data de naixement (DDMMAA).

ORIGEN: *Origen*. Camp de 5 dígitos alfanumèrics per indicar la granja de origen de l'animal. (Opcional).

FECE: *Fecha de Entrada en la Explotación*. 6 dígitos numèrics per fila data d'entrada en la explotació (DDMMAA).

BAJA: *Baja*. Camp d'una posició numèrica que s'activa o no depenent si l'animal esta o no donat de baixa.

0 Activo

1 Baja.

FECB: *Fecha de Baja.* Camp de 6 dígits numèrics que ens indica la data de baixa (DDMMAA).

RAZÓN : *Razón de la Baja.* Camp numèric de 2 posicions numèriques que indica el motiu de la baixa, segons sigui el codis.

Hembras.	Machos.
01 = no prenyada per cobriments	31 = muerte
02 = no apareix en zel	32 = infertilitat
03 = abortada	33 = subfertilitat (baix % fertilitat)
04 = baixa productivitat per la edat falta d'ardor	34 = dificultat acoblament,
05 = problemes de mal de peu	35 = camades petites
06 = morta en la explotació	36 = aploms
07 = baix nombre de nascuts	37 = coixeja o artritis
08 = baix nombre de destetats morts	38 = produeix garrins
09 = accidents al part	39 = vells y pesats
10 = trastorns en la lactació	40 = consanguinitat (tares,...)
11 = enfermetats respiratòries	41 = mal
12 = enfermetats declaració obligatòria	42 = ALTRES CAUSES

13 = evaginació de l'úter o del recte

14 = anomalies en garrins

15 = consanguinitat del ramat

16 = garrins nascuts amb poc pes

17 = mals canals en descendents

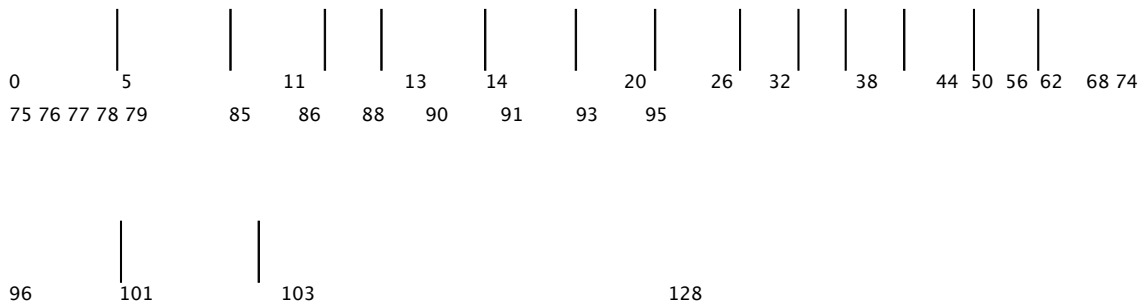
18 = males condicions de mercat

19 = venda com reproductors

20 = ALTRES CAUSES

REPRODUCTIVOS

Format VSI lineal



GRANJA: *Nº de Granja.* Camp 5 dígit alfanumèrics que representa la granja en que es troba l'animal.

ANIMAL: *Número del animal.* Camp de 6 dígit alfanumèrics que ens indica l'animal.

CICLO: *Número de ciclo.* Es el nombre de ciclo que se esta entrant. Camp de 2 dígit numèrics.

REP: *Número de repeticiones.* Son les repeticions que s'han realitzat en un mateix cicle. Camp de numèric 1dígit

FECHA CUB.: *Fecha de Cobrimentn.* Hi han 5 camps numèrics per la data de cobriment de 6 dígit, amb el que hi ha un màxim de 4 repeticions. Camp de 6 dígit numèrics que conté la data de cobriment (DDMMAA).

MACHO: *Número de Macho.* Hi han 5 camp alfanumèrics per la identificació d'un macho cada una correspon amb la data de cobriment. Cada camp compren 6 dígit alfanumèrics

TIPO DE CUBR: *Tipo de Cobrimentn.* Camp de 1 dígit numèric que agafa el codis del tipus de la cobriment, segons sigui el seu codis

1 = simple, només una monta.

2 = múltiples montes amb un sol macho

6 = inseminació artificial exterior.

8 = múltiples montes con machos diferents.

9 = inseminació artificial.

FECHA PARTO: *Fecha de Parto.* Camp de 6 dígit numèrics que conté la data de part (DDMMAA).

TIPO DE PARTO

0 ---> Part Normal

1 ---> Avortament

VIV: *Lechones Nacidos Vivos.* Camp de 2 dígits numèrics que indica el nombre de garrins (machos y hembras) nascuts vius.

MUE: *Lechones.* Camp numèric de 2 dígits que indica el nombre de nascuts morts.

MOMI: *Lechones Nacidos Momificados.* Camp numèric de un dígit que indica el nombre de momificats.

ADO: *Lechones Adoptados.* Camp numèric de 2 dígits que indica el nombre de garrins adoptats.

RET: *Lechones Retirados.* Camp numèric de 2 dígits que indica el nombre de garrins retirats.

FECHA DESTETE: *Fecha del Destete.* Camp de 6 dígits numèrics per la (DDMMAA).

Nº LECHONES: *Número de Lechones Destetados.* Camp numèric de 2 dígits que indica el nombre de garrins destetats

ANNEX III: MANUAL DE L'APLICACIÓ

ESTUDI I IMPLEMENTACIÓ D'UN SIMULADOR DE CICLE TANCAT DE PRODUCCIÓ PORCINA.

ÍNDEX

INTRODUCCIÓ 65

EL SIMULADOR 65

BLOCS DE PARÀMETRES 68

BLOCS DE D'ENTRADA I SORTIDA DE DADES 76

BLOCS DE TRANSICIÓ I ENGREIX 79

PROCÉS DE SIMULACIÓ 82

FORMAT DEL VECTOR ESTANDARD D'INFORMACIÓ TÈCNICA – V.S.I 85

INTRODUCCIÓ

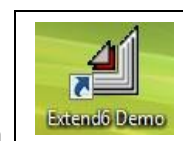
L'aplicació que es pretén utilitzar funciona a través del programari ExtendSim 6. L'aplicació serveix per calcular el comportament d'una

granja de garrins des del seu naixement fins a la seva matança a l'escorxador. També durant el seu cicle de vida, diferenciem entre si es tracta d'un garrí destinat a l'engreix o a la reproducció.

EL SIMULADOR

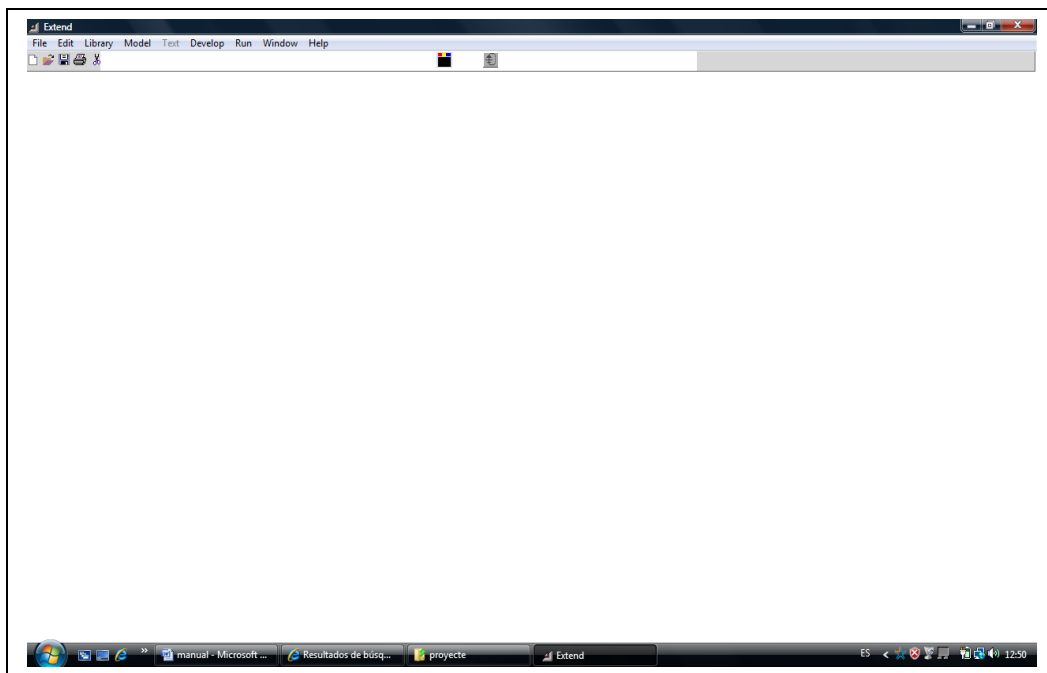
El primer que s'ha de fer per utilitzar l'aplicació és instal·lar el programari de l'ExtendSim 6. Aquest programa es pot obtenir a través de la pagina web www.ExtendSim.com que per tany a l'empresa Imagine That Inc..Aquest programa és el programari d'un simulador de esdeveniments discrets. Que sigui d'esdeveniments discrets significa que s'utilitza per simular fets que no ocorren de forma continua com podria ésser el moviment d'un planeta. Un exemple d'esdeveniment discret seria un sistema de pintura que només funciona quan detecta un cotxe al seu davant.

Un cop ens l'hem baixat e instal·lat podem començar a construir el nostre aplicació. N'hi ha prou amb l'ExtendSim, el model i un conjunt de llibreries. El model que utilitzarem és el Model2008e. Les llibreries que hauríem de tenir guardades a l'escriptori són : FarmSim2008 i SowFarmBasic. Tots ells el guardarem a l'escriptori.

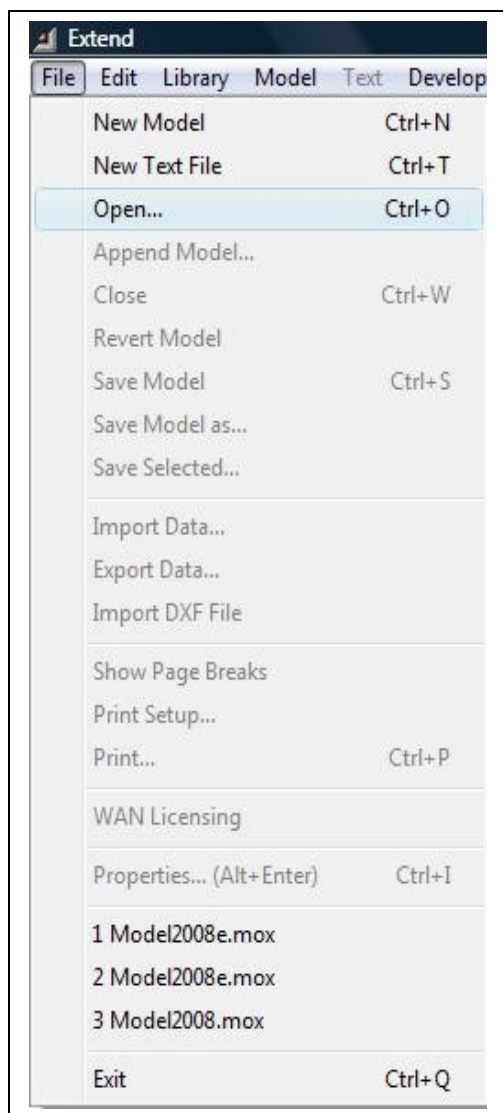


Quan obrim l'ExtendSim 6 fent un doble clic a l'icona s'obrirà el programa i visualitzarem el programa. El resultat final és:

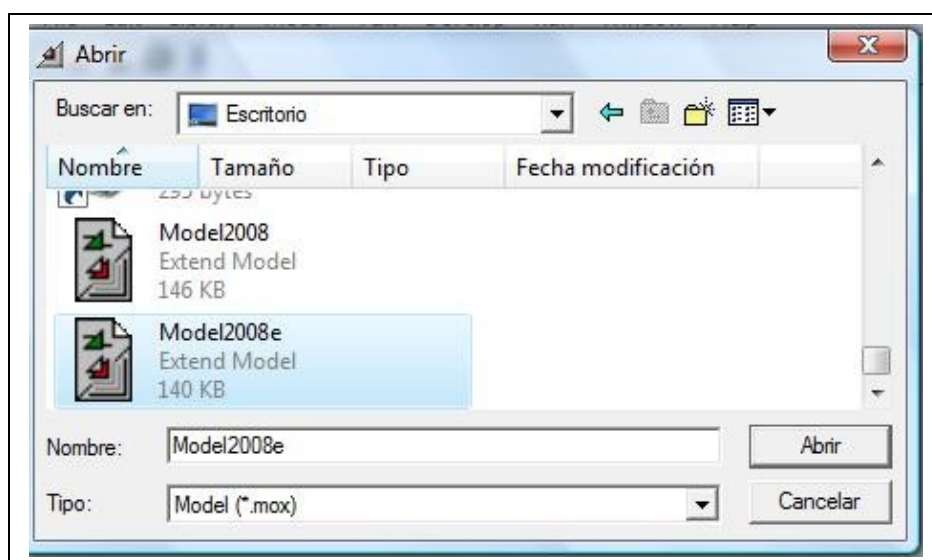
Estudi i implementació d'un simulador de cicle tancat de producció de porcs.



Si volem veure el nostre model, el que farem a continuació és clicar el menú File que es troba a la barra de menús que és la segona línia que apareix a la imatge anterior. Se'ns obrirà la finestra següent i escollirem l'opció Open:

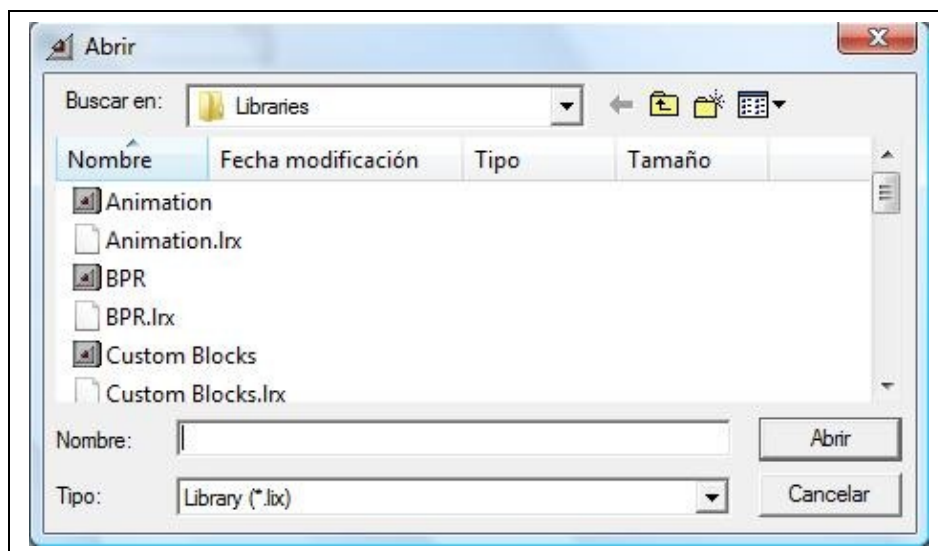


Se'ns obrirà la finestra següent:

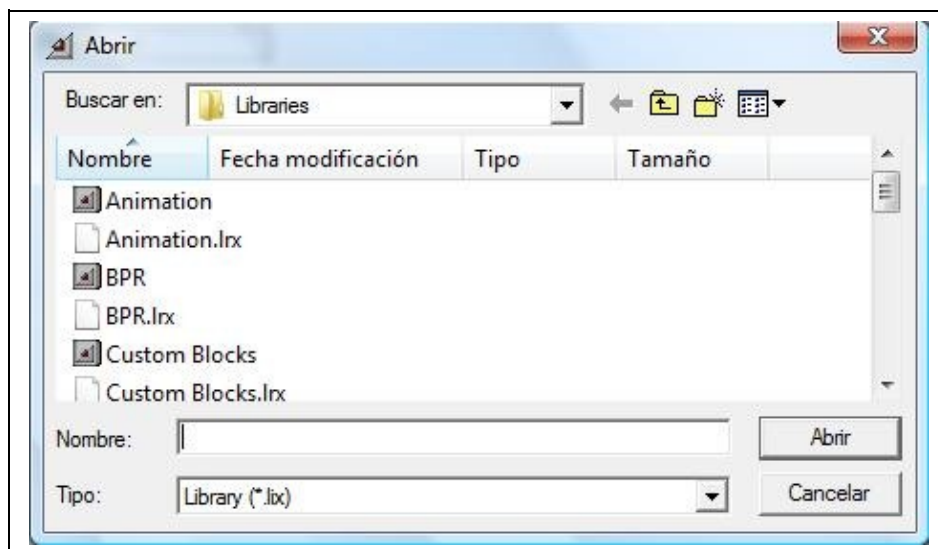


Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

En Buscar en: farem un clic a la fletxa de la dreta del quadre de text , se'ns obrirà un llistat i escollirem l'Escriptorio. En el quadre de text de Nombre: escriurem Model2008e i farem un clic en Abrir. Ara se'ns obrirà la finestra següent:

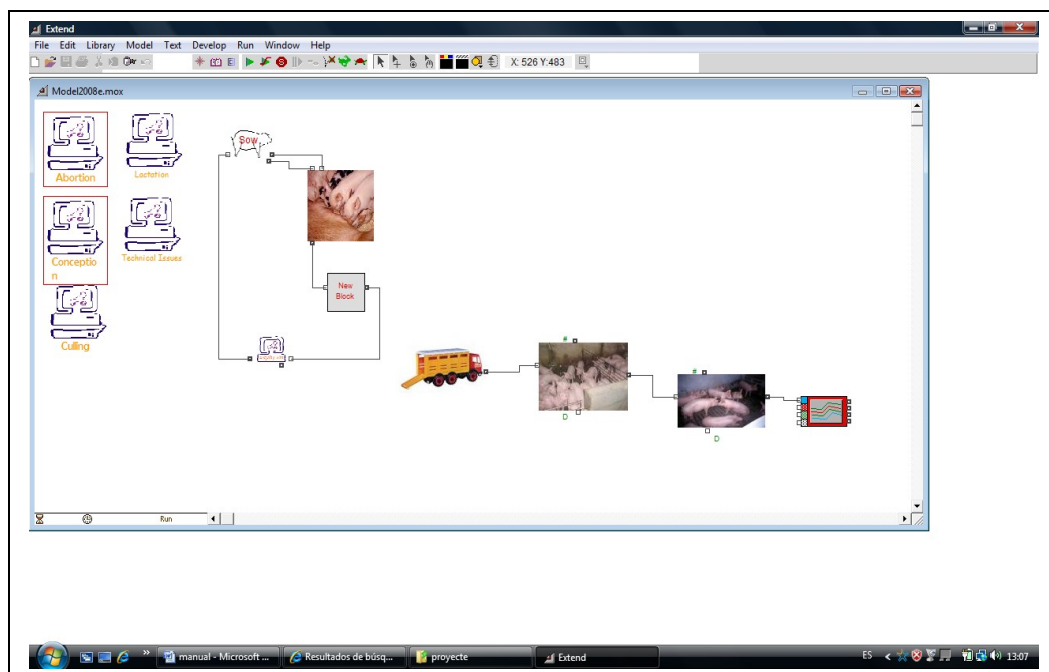


En Buscar en: farem un clic a la fletxa del quadre de text , se'ns obrirà un llistat i escollirem l'Escriptorio. En el quadre de text de Nombre: escriurem FarmSim2008 i farem un clic en Abrir. Ara se'ns obrirà la finestra següent:



En Buscar en: farem un clic a la fletxa del quadre de text , se'ns obrirà un llistat i escollirem l'Escriptorio. En el quadre de text de Nombre: escriurem SowFarmBasic i farem un clic en Abrir. Ara se'ns obrirà la finestra següent:

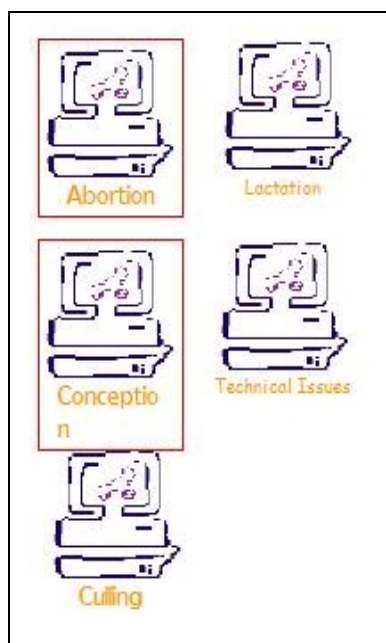
Estudi i implementació d'un simulador de cicle tancat de producció de porcs.



En aquest moment ja hem instal·lat el model al programa ExtendSim 6.

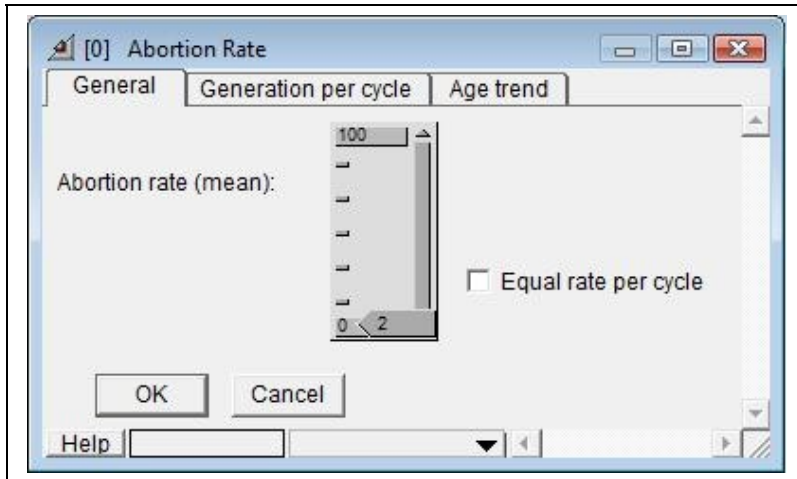
BLOCS DE PARÀMETRES

Ara podem posar a prova el nostre simulador. A l'esquerra del nostre model estan els blocs de paràmetres com apareix a l'imatge següent:



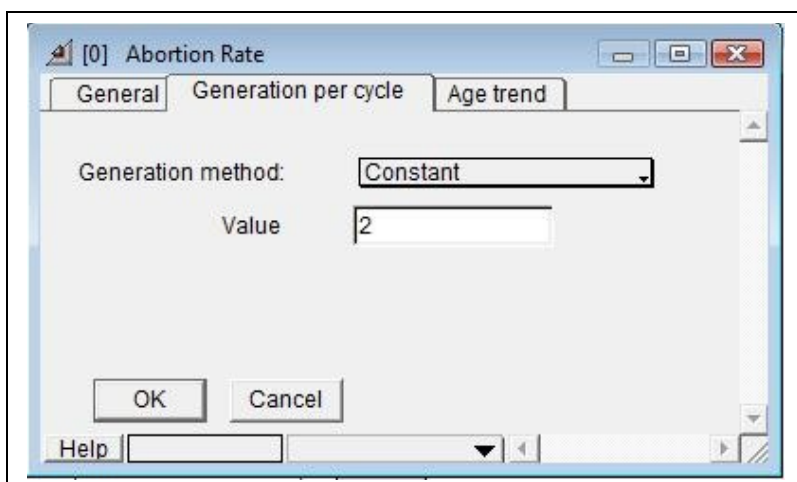
Bloc abortion

Al fer un doble clic damunt de cadascun d'ells obtenim el valor dels seus paràmetres que podem modificar. Primer fem un doble clic damunt el bloc Abortion ens apareix la següent figura:



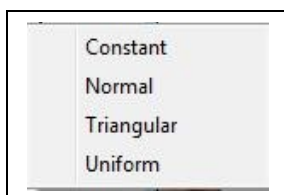
En la pestanya General podem canviar el Abortion rate (mean) o mitjana del radi d'avortament fent un clic damunt de la figura vertical. El seu valor va de 0 a 100. Si fem un clic damunt la fletxa amb el nombre 2 i sense prémer la desplaçem cap a dalt modificarem el seu valor. Si fem un clic damunt el rectangle de Equal rate per cycle li direm que el cicle del radi es igual.

Si cliquem a la pestanya Generation per cycle se'ns obrirà la figura següent:



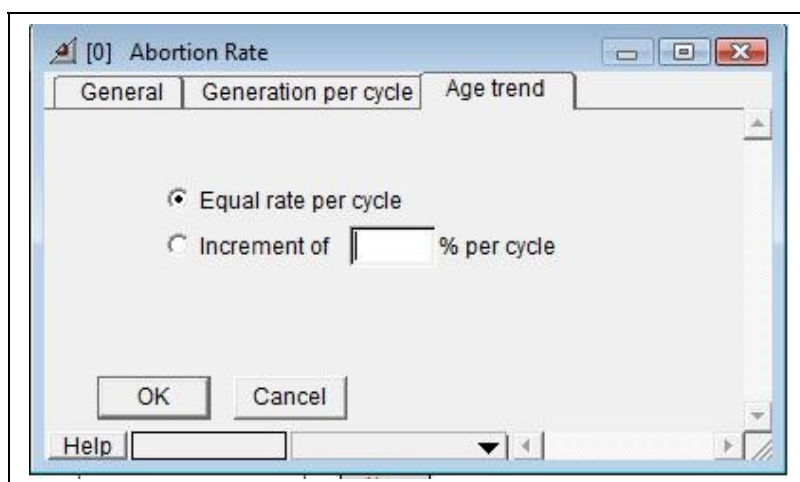
En la pestanya Generation per cycle podem canviar el Generation method o mètode de generació fent un clic damunt de la fletxa i escollint una

opció de la llista vertical. Les opcions que poden ésser son



Podem modificar el valor de value fent un clic damunt de la caixa de text , esborrant i canviant el nombre.

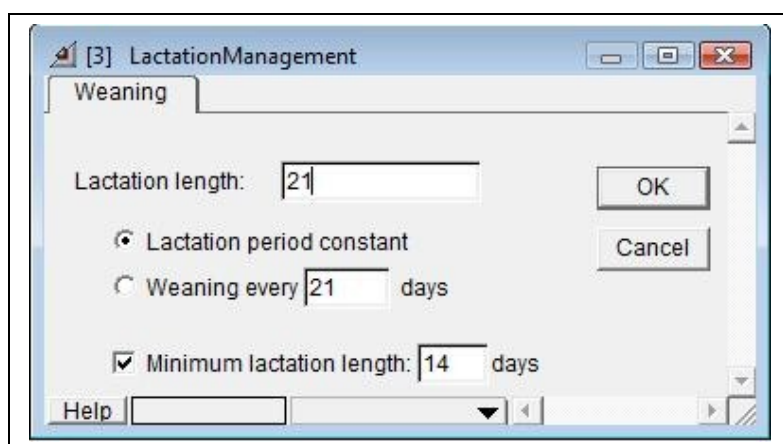
Si cliquem a la pestanya Age trend se'ns obrirà la figura següent:



En la pestanya Age trend podem canviar el Increment of % per cycle o increment d'un % per cicle fent un clic damunt d'aquesta opció i un clic en el quadre de text i col·locant un nombre. Per tenir un radi igual per cicle triem l'opció Equal rate per cycle.

Bloc lactation

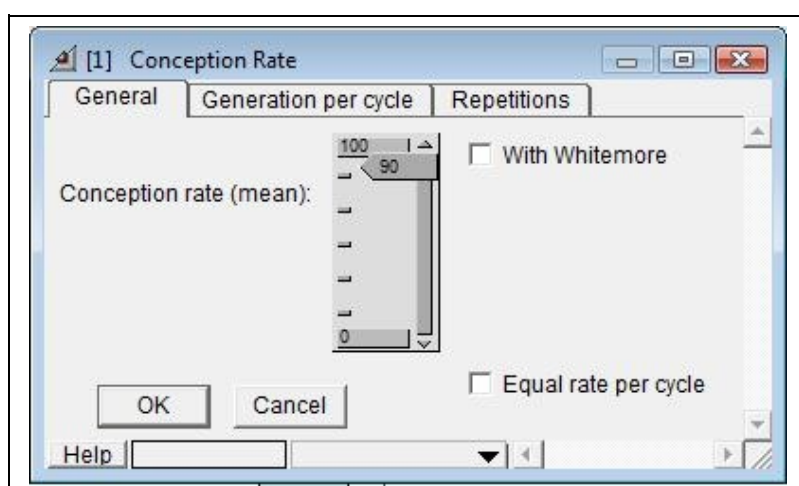
Ara fem un doble clic damunt el bloc Lactation ens apareix la següent figura:



En la pestanya Weaning podem canviar el Lactation length o durada de lactació fent un clic damunt el quadre de text i col·locant un nombre. Per tenir una lactació de període constant triem l'opció Lactation period constant. Per indicar un desllet cada x dies fem un clic a l'opció Weaning every days i un clic al quadre de text posant un nombre. Finalment indicarem el període mínim de la durada de lactació fent un clic a l'opció Minimum lactation length: days i un clic al quadre de text posant un nombre. Per acabar cliquem al botó ok.

Bloc conception

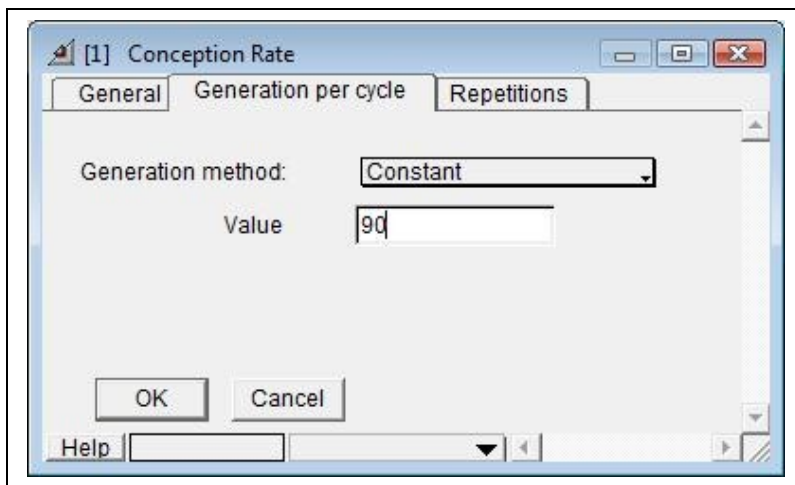
Ara fem un doble clic damunt el bloc conception ens apareix la següent figura:



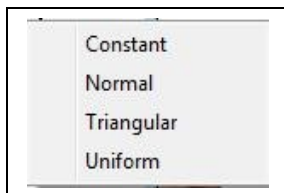
En la pestanya General podem canviar el Conception rate (mean) o mitjana del radi de concepció fent un clic damunt de la figura vertical. El seu valor va de 0 a 100. Si fem un

clic damunt la fletxa amb el nombre 2 i sense prémer la desplaçem cap a dalt modificarem el seu valor. Si fem un clic damunt el rectangle de Equal rate per cycle li direm que el cicle del radi es igual. Si fem un clic a l'opció Whitemore calcularà la formula 1998 pp439: $\Delta WG = 25FG(kg) - 27$.

Si cliquem a la pestanya Generation per cycle se'ns obrirà la figura següent:

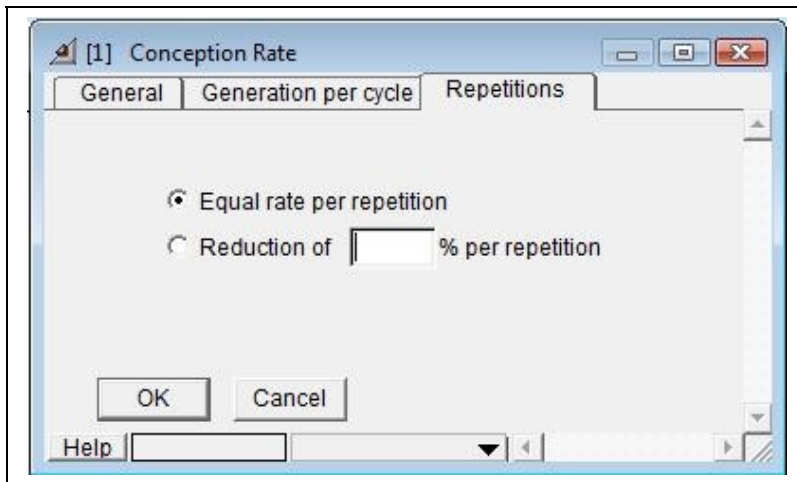


En la pestanya Generation per cycle podem canviar el Generation method o mètode de generació fent un clic damunt de la fletxa i escollint una opció de la llista vertical. Les opcions que poden ésser son



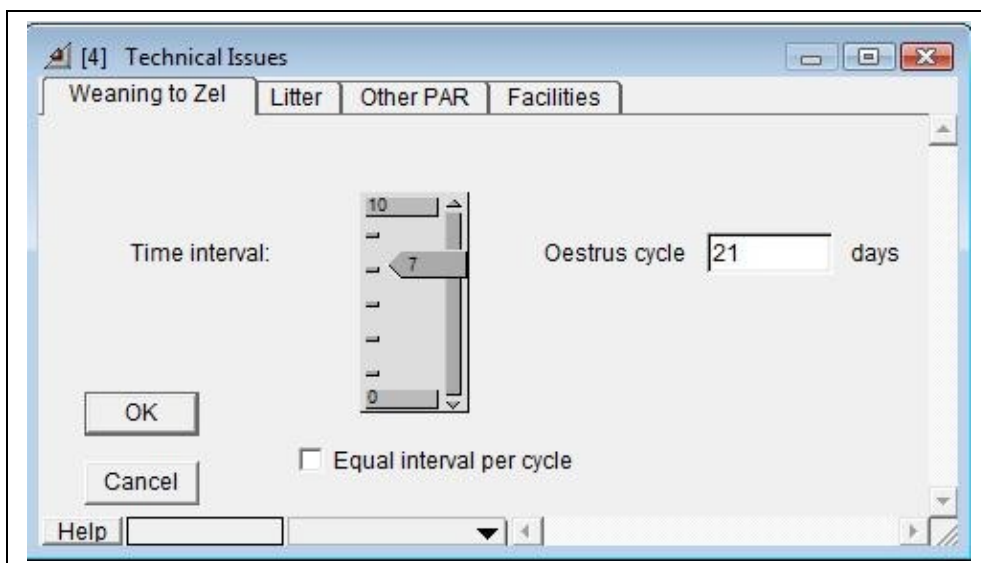
Podem modificar el valor de value fent un clic damunt de la caixa de text , esborrant i canviant el nombre.

Si cliquem a la pestanya Repetitions se'ns obrirà la figura següent:



En la pestanya Repetitions podem canviar el Reduction of % per repetition o decrement d'un % per repetició fent un clic damunt d'aquesta opció i un clic en el quadre de text i col·locant un nombre. Per tenir un radi igual per repetició triem l'opció Equal rate per repetition.

Ara fem un doble clic damunt el bloc Technical Issues ens apareix la següent figura:

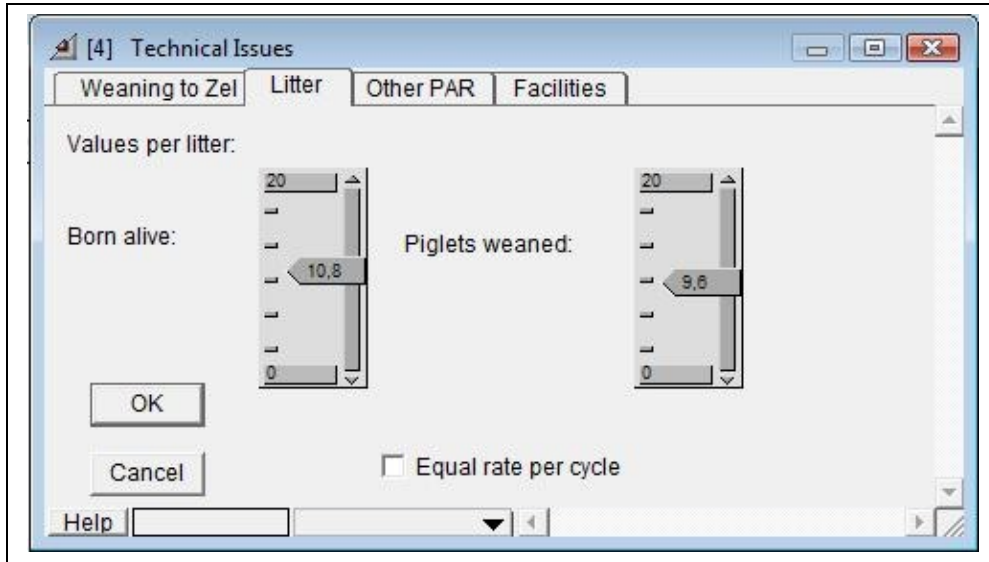


En la pestanya Weaning to Zel podem canviar el Time interval o l'interval de temps fent un clic damunt de la figura vertical. El seu valor va de 0 a 100. Si fem un clic damunt la fletxa amb el nombre 7 i sense prémer la desplaçem cap a dalt modificarem el seu valor. Si fem un clic damunt el rectangle de Equal rate per cycle li direm que el cicle del radi

es igual. Si fem un clic damunt del quadre de text Oestrus cycle days canviarem el cicle de dies del zel.

Bloc technical issues

Si cliquem a la pestanya Litter se'ns obrirà la figura següent:



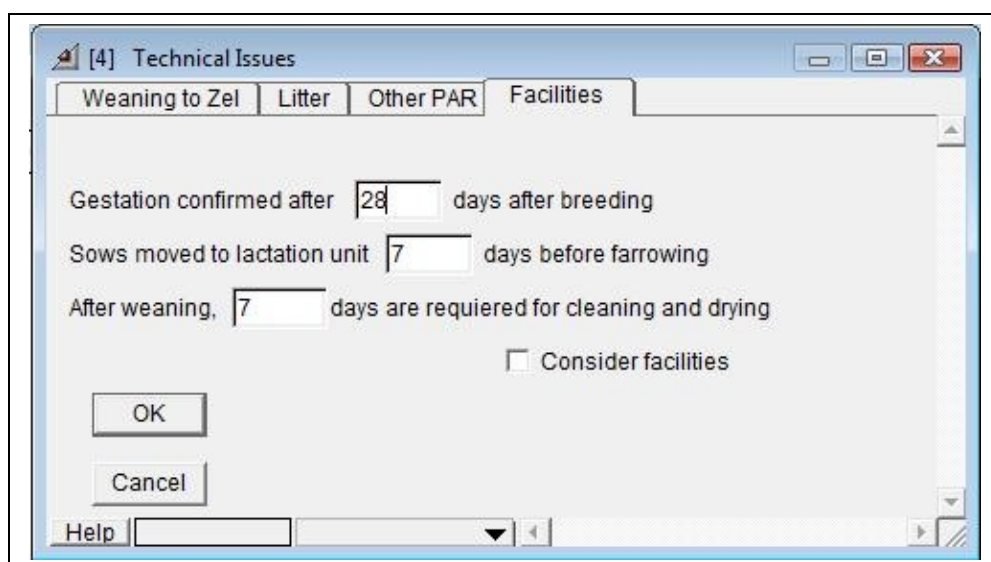
En la pestanya Litter podem canviar el Born alive o nascuts vius fent un clic damunt de la figura vertical. El seu valor va de 0 a 20. Si fem un clic damunt la fletxa amb el nombre 10,8 i sense prémer la desplaçem cap a dalt modificarem el seu valor. Podem canviar el Piglets weaned o garrins deslletats fent un clic damunt de la segona figura vertical. El seu valor va de 0 a 20. Si fem un clic damunt la fletxa amb el nombre 9,6 i sense prémer la desplaçem cap a dalt modificarem el seu valor. Si fem un clic damunt el rectangle de Equal rate per cycle li direm que el cicle del radi es igual.

Si cliquem a la pestanya Other PAR se'ns obrirà la figura següent:



En la pestanya Other PAR podem canviar el valor de Age of 1st mating days o edat del primer muntatge fent un clic damunt el quadre de text i canviant el valor. Podem canviar el valor de Gestation length: days o durada d'embaràs fent un clic damunt el quadre de text i canviant el valor. Podem canviar el valor de Abortion: days o durada d'avortament fent un clic damunt el quadre de text i canviant el valor.

Si cliquem a la pestanya Facilities se'ns obrirà la figura següent:

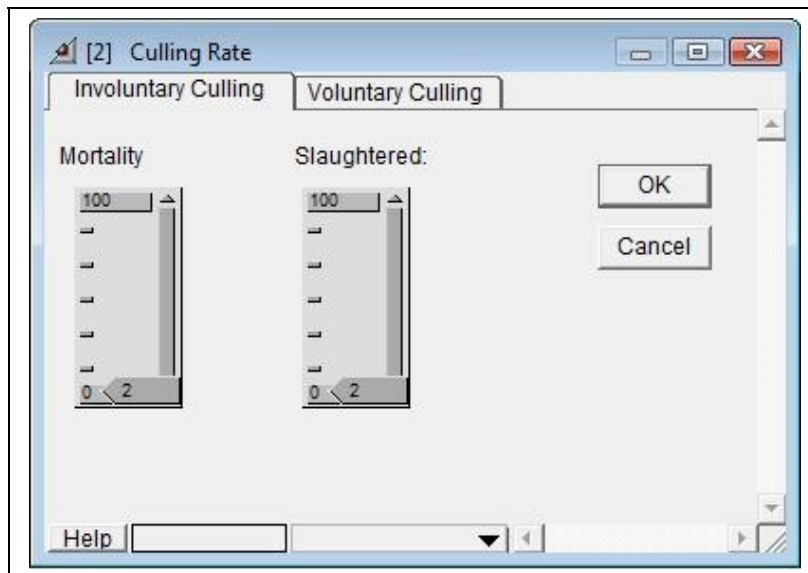


En la pestanya Facilities podem canviar el valor de Gestation confirmed after days of breeding o embaràs confirmat després de dies d'alimentar-se fent un clic damunt el

quadre de text i canviant el valor. Podem canviar el valor Sows moved to lactation unit days before farrowing o truïes mogudes a la unitat de lactació fent un clic damunt el quadre de text i canviant el valor. Podem canviar el valor de After weaning, days are required for cleaning and drying o durada requerida per neteja i assecat després de desllet fent un clic damunt el quadre de text i canviant el valor. Finalment fem un clic al botó ok.

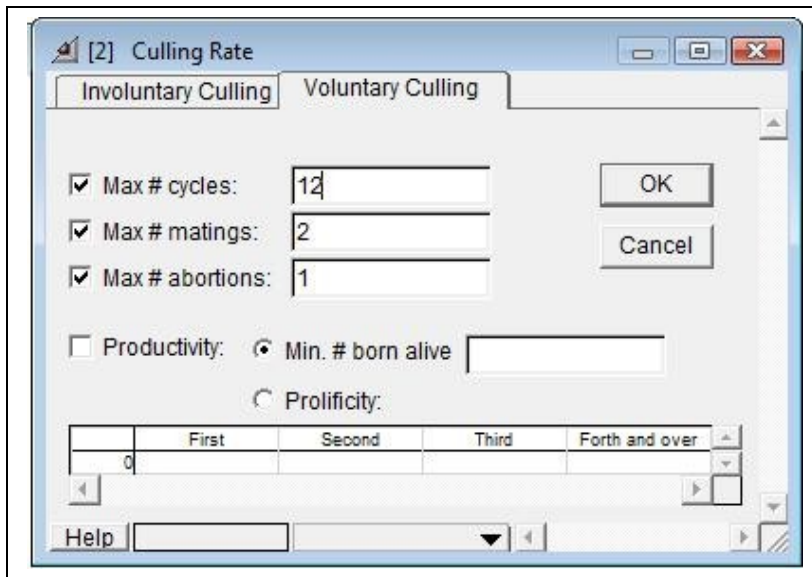
Bloc culling:

Ara fem un doble clic damunt el bloc Culling ens apareix la següent figura:



En la pestanya Involuntary Culling podem canviar el Mortality o mortalitat fent un clic damunt de la figura vertical. El seu valor va de 0 a 100. Si fem un clic damunt la fletxa amb el nombre 2 i sense prémer la desplaçem cap a dalt modificarem el seu valor. Podem canviar el Slaughtered o sacrificats fent un clic damunt de la segona figura vertical. El seu valor va de 0 a 100. Si fem un clic damunt la fletxa amb el nombre 2 i sense prémer la desplaçem cap a dalt modificarem el seu valor.

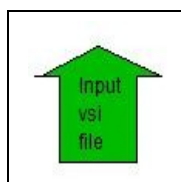
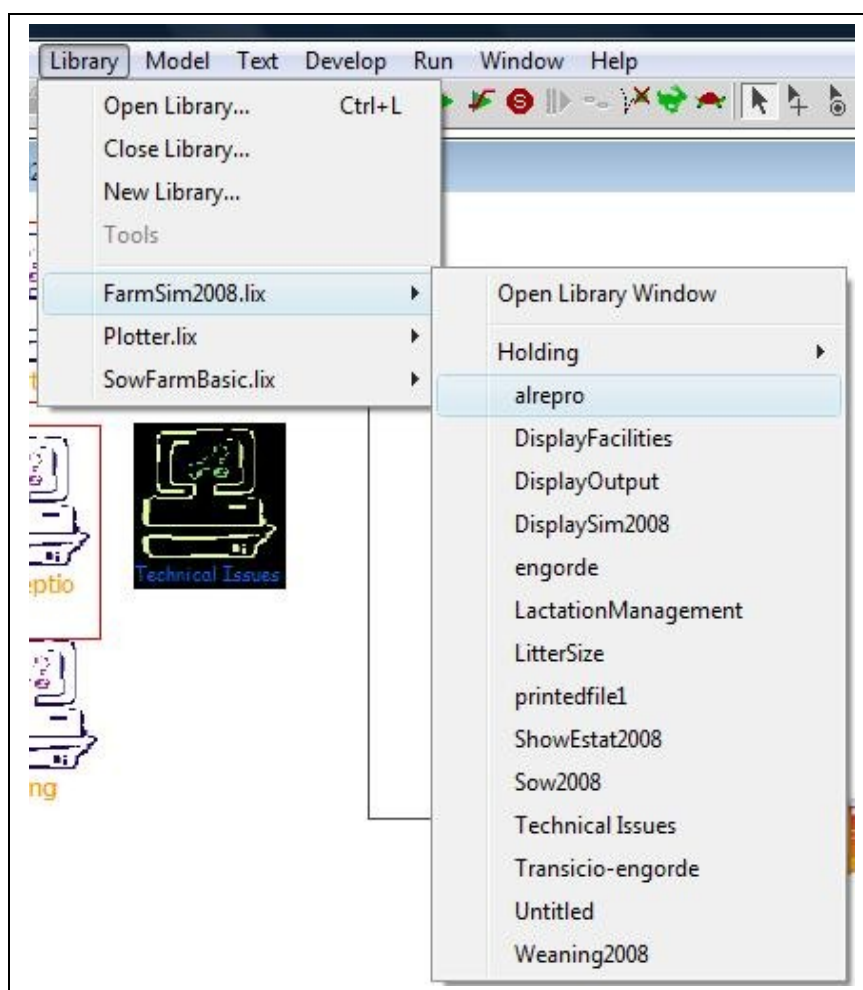
Si cliquem a la pestanya Voluntary Culling se'ns obrirà la figura següent:



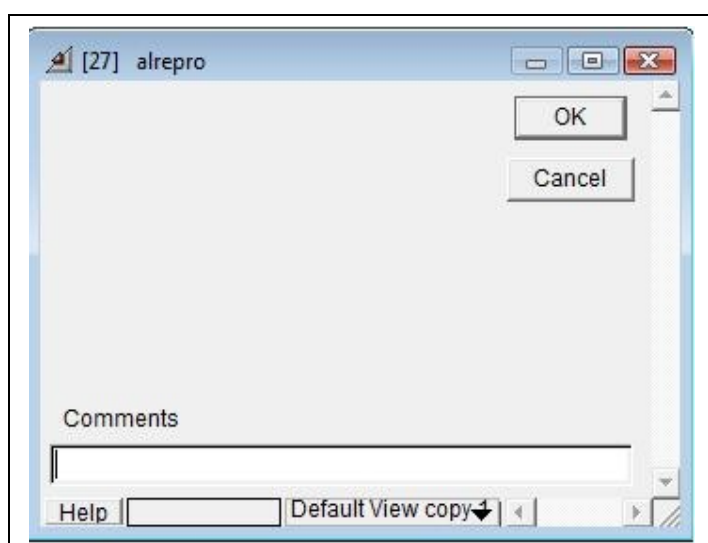
En la pestanya Voluntary Culling podem canviar el valor de Max#cycles o nombre màxim de cicles fent un clic damunt el quadre de text i canviant el valor. Podem canviar el valor Max#matings o nombre màxim de muntatges fent un clic damunt el quadre de text i canviant el valor. Podem canviar el valor de Max#abortions o durada nombre màxim d'avortaments fent un clic damunt el quadre de text i canviant el valor. Podem clicar l'opció de Min#born alive o mín de nascuts vius fent canviant el valor al quadre de text. Fent un clic a l'opció Prolificity fas l'opció de prolificitat. Finalment fem un clic al botó ok.

BLOCS D'ENTRADA I SORTIDA DE DADES

Aquest valors introduïts a mà els podem posar de forma automàtica des d'un fitxer d'extensió .vsi o fitxer Excel des d'un disquet extern a través del bloc d'entrada de dades o input file vsi. En el nostre cas el fitxer es trobarà en la unitat c:\. La forma d'obtenir-lo és fent un clic a l'opció Library de la barra de menús. Després es desplegarà diferents opcions i escollirem el bloc alrepro de la llibreria FarmSim2008 com es veu en la figura següent:

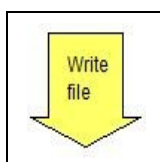
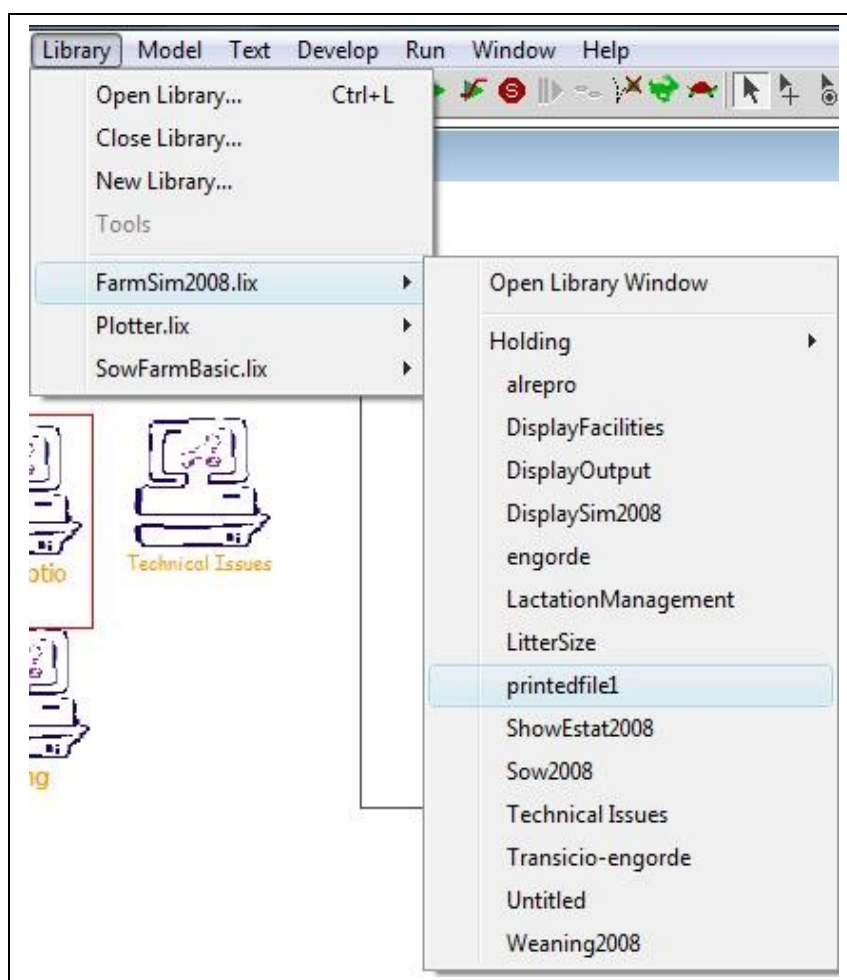


Al clicar es crearà el bloc . Al fer doble clic a la figura apareixerà la figura següent:

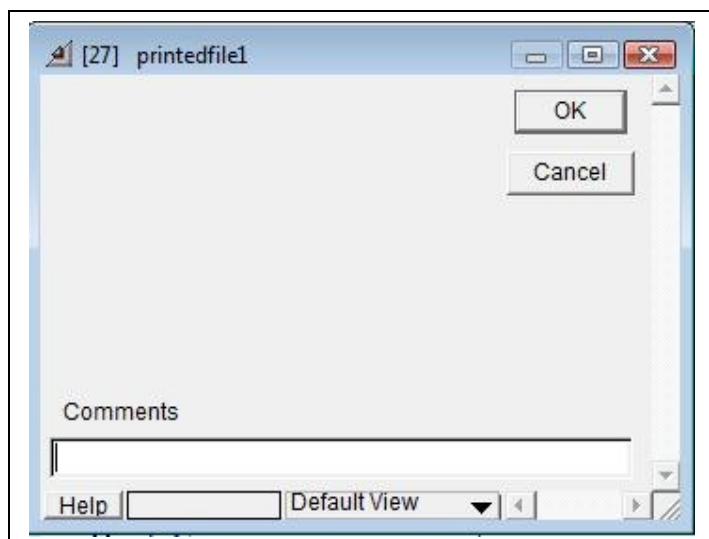


Per acabar s'ha de fer un clic a ok.

En canvi, si el que volem es posar totes les dades dels blocs de paràmetres a un fitxer en la unitat c: hem de posar el bloc de sortida de dades. Ara pretenem posar aquest bloc al model. La forma d'obtenir-lo és fent un clic a l'opció Library de la barra de menús. Després es desplegarà diferents opcions i escollirem el bloc printedfile1 de la llibreria FarmSim2008 com es veu en la figura següent:



Al clicar es crearà el bloc . Al fer doble clic a la figura apareixerà la figura següent:



Per acabar s'ha de fer un clic a ok.

BLOCS DE TRANSICIÓ I ENGREIX

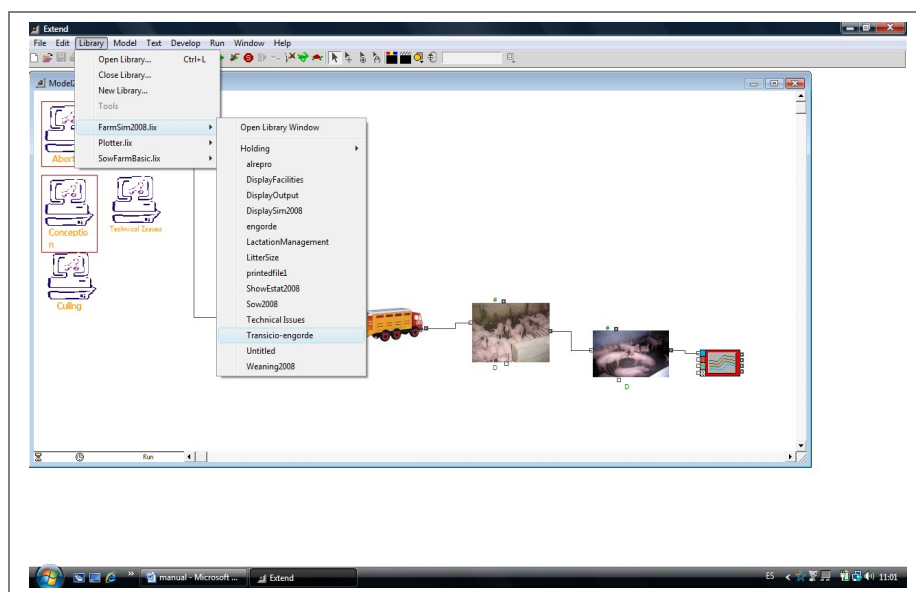
Per poder introduir en el simulador els blocs de transició i engreix hem de seguir els passos següents:

1º-Buscar el bloc de transició de la forma següent:

1.1º-Fer un clic a l'opció Library de la barra de menús.

1.2º-Fer un clic a l'opció FarmSim2008

1.3º-Fer un clic damunt del bloc Transicio-engorde



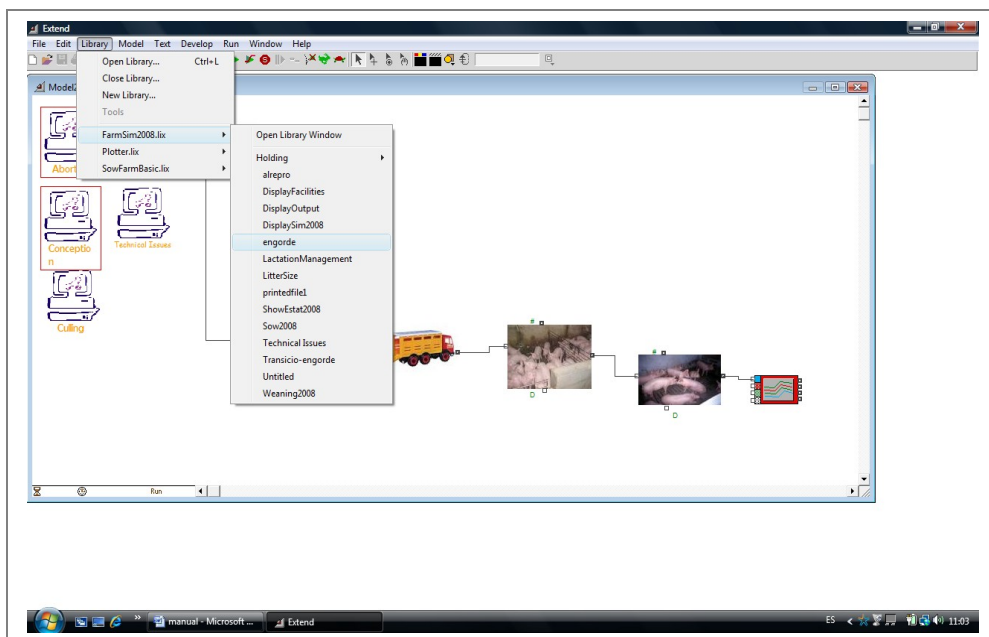
Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

2º-Buscar el bloc d'engreix de la forma següent:

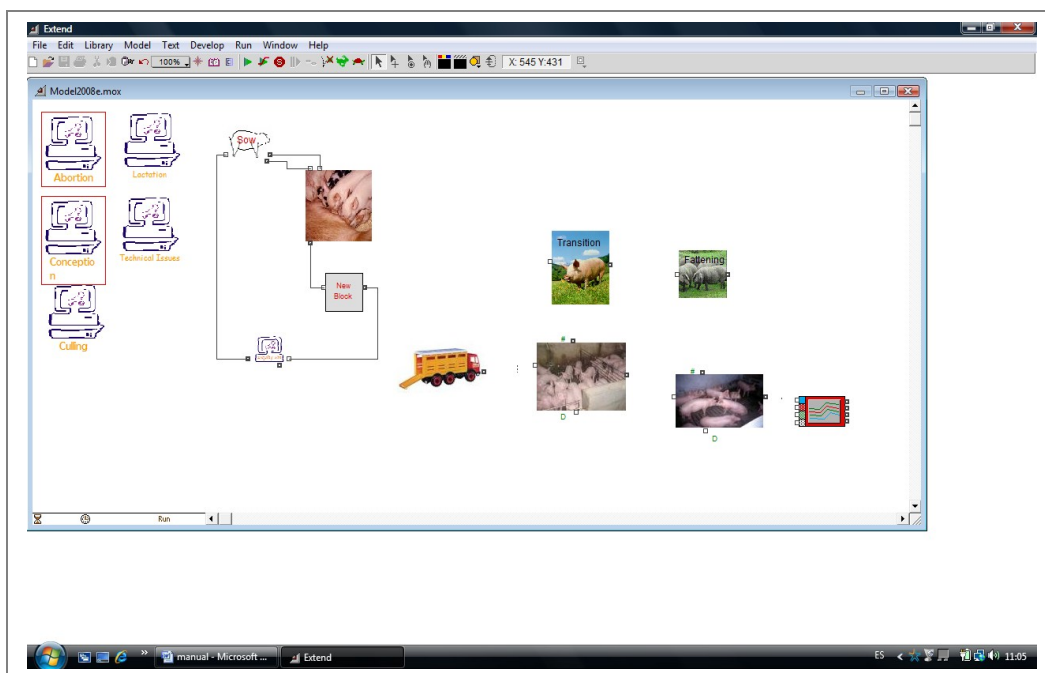
2.1º Fer un clic a l'opció Library de la barra de menús.

2.2º-Fer un clic a l'opció FarmSim2008

2.3º-Fer un clic damunt del bloc engorde

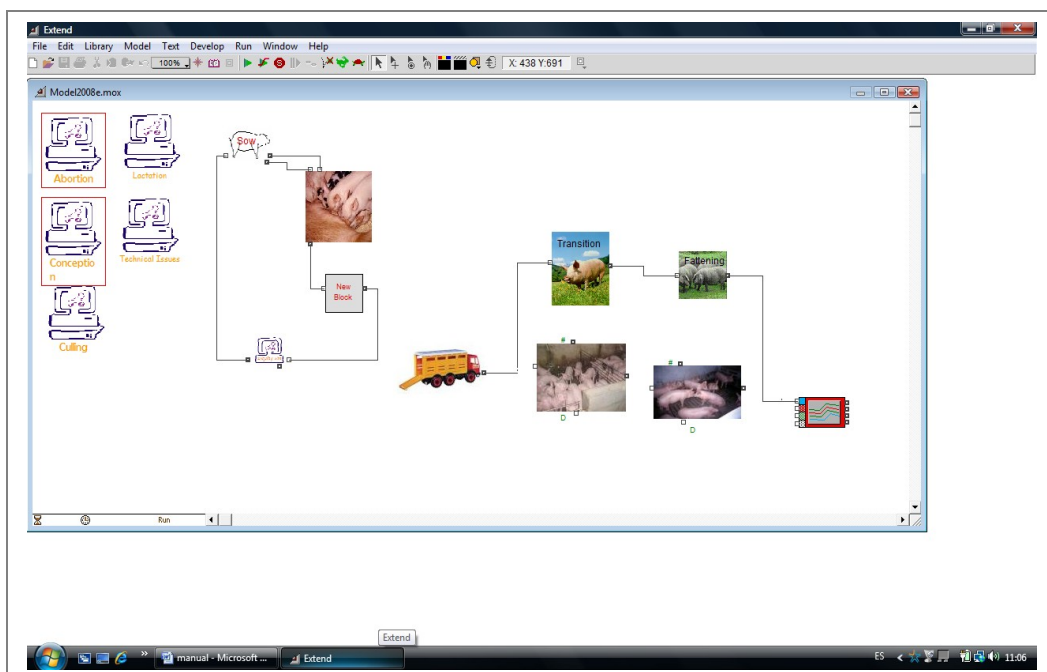


3º-Fer un clic damunt les línies que uneixen el camió del simulador amb els blocs posteriors i el visualitzador.

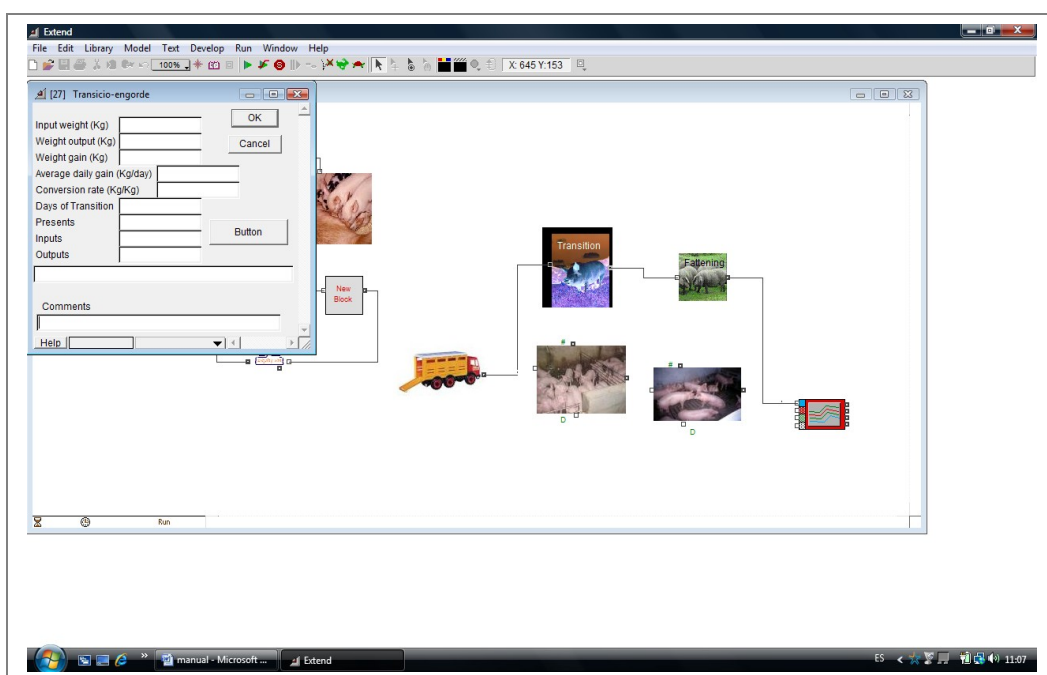


Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

4º-Fer un clic des del connector del camió al bloc transició-engreix, un altre clic, mantenint pres, des del connector de sortida del bloc transició-engreix al connector d'entrada del bloc engorde i finalment del connector de sortida del bloc engorde al connector del bloc de visualització.

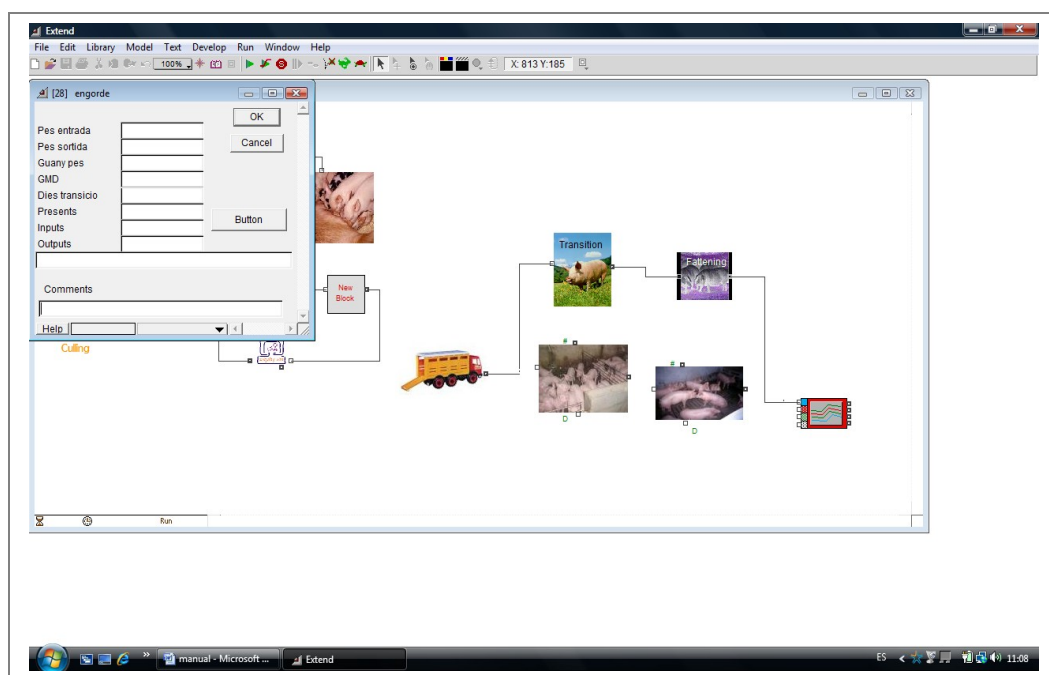



5º-Fer un doble clic al bloc de transició-engorde i omplir les dades fent un clic damunt dels quadres de text, del botó Button i fent clic al botó ok.



Estudi i implementació d'un simulador de cicle tancat de producció de porcs.

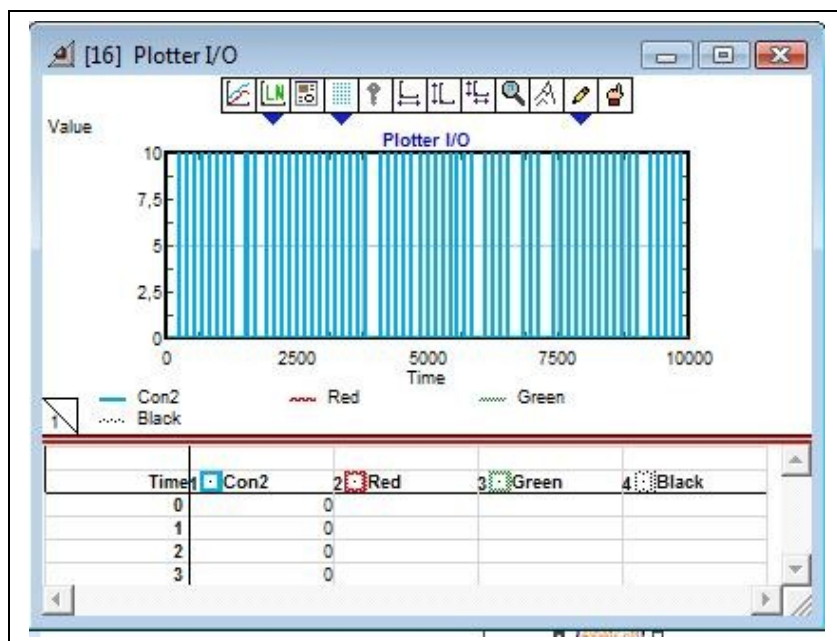
6º- Fer un doble clic al bloc de engorde i omplir les dades fent un clic damunt dels quadres de text, del botó Button i fent clic al botó ok.



7º-Fer un clic al botó Run  de la barra de botons, que es troba davall de la barra de menús.

PROCÉS DE SIMULACIÓ

S'obrirà el bloc de visualització dels resultats, amb unes dades determinades d'exemple es veuria la figura següent:



Dono per finalitzat aquest manual.

Afegim el format que han de tenir els fitxers vsi d'entrada de dades:

FORMAT DEL VECTOR ESTANDARD D'INFORMACIÓ TÈCNICA - V.S.I.-

ALTAS

Formato VSI lineal

GRANJA	ANIMA	CICL	SEX	RAZ	FECN	ORIGE	FECA	BAJA	FECB	RAZÓN	BLANCOS
	L	O	O	A		N					
0	5		11	13		14	16	22		27	33
40	42			128							34

GRANJA: *Nº de Granja.* És un camp de 5 dígits que ens indica el nombre de granja

ANIMAL: *Nº de Animal* Es un camp de 6 dígits, distribuïts en,

Año: Última xifra de l'any de la data de naixement (1 dígit)

Orden: En el cas de que un nombre d'animal es repetís el nombre de ordre seria diferent, sinó por defecte se posaria el 0 (1 dígit)

Nº del animal : Nombre d' identificació del reproductor. (4 dígits)

CICLO: *Nº de ciclo.* Ens indica el nombre de cicle en el que es troba l'animal en el moment de la alta, normalment es 00. (2 dígits)

SEXO: *Sexo del Animal.* Es un nombre identificador del sexe del animal. (1 dígit)

1 hembra.

2 Macho.

RAZA: *Raza del Animal.* Camp de 2 dígit numèrics que indica la raça del **pare** y la **mare** del reproductor en aquest ordre.

1 = Large White.

2 = Landrace "materno".

3 = Landrace Alemán.

4 = Landrace Belga.

5 = Duroc.

6 = Pietrain.

7 = Hampshire.

8 = (altra raça).

9 = animal cruzado.

FECN: *Fecha de Nacimiento.* Camp de 6 dígit numèrics para la data de naixement (DDMMAA).

ORIGEN: *Origen.* Camp de 5 dígit alfanumèrics per indicar la granja de origen de l'animal. (Opcional).

FECE: *Fecha de Entrada en la Explotación.* 6 dígit numèrics per la data d'entrada en la explotació (DDMMAA).

BAJA: *Baja.* Camp d'una posició numèrica que s'activa o no depenent si l'animal esta o no donat de baja.

0 Activo

1 Baja.

FECB: *Fecha de Baja.* Camp de 6 dígitos numèrics que ens indica la data de baixa (DDMMAA).

RAZÓN : *Razón de la Baja.* Camp numèric de 2 posicions numèriques que indica el motiu de la baixa, segons sigui el codis.

Hembras.	Machos.
01 = no prenyada tras cubricions	31 = muerte
02 = no apareix en zel	32 = infertilitat
03 = avortada	33 = subfertilitat (baix % fertilitat)
04 = baixa productivitat per la edat falta d'ardor	34 = dificultat acoblament,
05 = problemes de mal de peu	35 = camades petites
06 = morta en la explotació	36 = aploms
07 = baix nombre de nascuts	37 = coixeja o artritis
08 = baix nombre de destetats morts	38 = produeix garrins
09 = accidents al part	39 = vells y pesats
10 = trastorns en la lactació	40 = consanguinitat (tares,...)

- 11** = enfermetats respiratòries **41** = mal
- 12** = enfermetats declaració obligatòria **42** = ALTRES CAUSES
- 13** = evaginació de l'úter o del recte
- 14** = anomalies en garrins
- 15** = consanguinitat del ramat
- 16** = garrins nascuts amb poc pes
- 17** = mals canals en descendents
- 18** = males condicions de mercat
- 19** = venta com reproductors **20** = ALTRES CAUSES

REPRODUCTIVOS

Formato VSI lineal

GRANJA	ANIMAL	CICLO	REP	FECHA CUB.	MACHO	TIPO CUBR	FECHA PARTO	TIP O	VI V	MUE	MOMI	ADO	RET
0 75 76	5	11		13	14	20	26	32		38	44	50 56	62 68 74
FECHA DESTETE	N° LECHONES	ESPACIOS EN BLANCO											
96	101	103					128						

GRANJA: *Nº de Granja.* Camp 5 dígit alfanuméric que representa la granja en que es troba l'animal.

ANIMAL: *Número del animal.* Camp de 6 dígit alfanuméric que ens indica l'animal.

CICLO: *Número de ciclo.* Es el nombre de ciclo que se esta entrant. Camp de 2 dígit numèrics.

REP: *Número de repeticions.* Son les repeticions que s'han realitzat en un mateix cicle. Camp de numèric 1dígit

FECHA CUB.: *Fecha de Cubrición.* Hi han 5 camps numèrics per la data de cubrició de 6 dígit, amb el que hi ha un màxim de 4 repeticions. Camp de 6 dígit numèrics que conté la data de cubrició (DDMMAA).

MACHO: *Número de Macho.* Hi han 5 camp alfanumèrics per la identificació d'un macho cada una correspon amb la data de cubrició. Cada camp compren 6 dígit alfanumèrics

TIPO DE CUBR: *Tipo de Cubrición.* Camp de 1 dígit numèric que agafa el codis del tipus de la cubrició, segons sigui el seu codis

1 = simple, només una monta.

2 = múltiples montes amb un sol macho

6 = inseminació artificial exterior.

8 = múltiples montes con machos diferents.

9 = inseminació artificial.

FECHA PARTO: *Fecha de Parto.* Camp de 6 dígit numèrics que conté la data de part (DDMMAA).

TIPO DE PARTO

0 ---> Part Normal

1 ---> Avortament

VIV: *Lechones Nacidos Vivos.* Camp de 2 dígits numèrics que indica el nombre de garrins (machos y hembras) nascuts vius.

MUE: *Lechones.* Camp numèric de 2 dígits que indica el nombre de nascuts morts.

MOMI: *Lechones Nacidos Momificados.* Camp numèric de un dígit que indica el nombre de momificats.

ADO: *Lechones Adoptados.* Camp numèric de 2 dígits que indica el nombre de garrins adoptats.

RET: *Lechones Retirados.* Camp numèric de 2 dígits que indica el nombre de garrins retirats.

FECHA DESTETE: *Fecha del Destete.* Camp de 6 dígits numèrics per la (DDMMAA).

Nº LECHONES: *Número de Lechones Destetados.* Camp numèric de 2 dígits que indica el nombre de garrins destetats

AGRAÏMENTS

A tot el que m'hagi ajudat i a la meva família.